Optimization of Robot Motion Planning using Ant Colony Optimization

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

Master of Technology

in

Production Engineering

2009-2011

by

Sangita Sarangi

(Roll no. 209ME2206)

Department of Mechanical Engineering
National Institute of Technology
Rourkela 769008, Orissa, India

May-2011

National Institute of Technology, Rourkela

CERTIFICATE

This is to certify that thesis entitled, **"Optimization of Robot Motion Planning using Ant Colony Optimization"** submitted by **Ms. Sangita Sarangi** in partial fulfillment of the requirements for the award of Master of Technology Degree in Mechanical Engineering with specialization in "**Production Engineering**" at National Institute of Technology, Rourkela is an authentic work carried out by her under my supervision and guidance.

To the best of my knowledge, the matter embodied in this report has not been submitted to any other university/ institute for award of any Degree or Diploma.

She bears a good moral character to the best of my knowledge and belief.

Place: NIT Rourkela
Date:

Dr. B.B. BISWAL
Dept. of Mechanical Engineering
National Institute of Technology
Rourkela-769008

ACKNOWLEDGEMENT

It is with a feeling of great pleasure that I would like to express my most sincere heartfelt gratitude to **Prof. B.B. Biswal**, Professor, Dept. of Mechanical Engineering, NIT Rourkela for suggesting the topic for my thesis report and for his ready and able guidance throughout the course of my preparing the report. I am greatly indebted to him for his constructive suggestions and criticism from time to time during the course of progress of my work.

I express my sincere thanks to **Prof. R.K.Sahoo**, Head of the Department of Mechanical Engineering, NIT, and Rourkela for providing me the necessary facilities in the department. I express my sincere gratitude to **Prof. S.S. Mahapatra**, Coordinator of M.Tech course for his timely help during the course of work. I am also thankful to all the staff members of the department of Mechanical Engineering and to all my well wishers for their inspiration and help. I feel pleased and privileged to fulfill my parent's ambition and I feel pleased and privileged to fulfill my parent's ambition and I am greatly indebted to them for bearing the inconvenience during my M.Tech course.

Date: SANGITA SARANGI

Roll No.:-209 ME 2206

ASTRACT

Motion planning in robotics is a process to compute a collision free path between the initial and final configuration among obstacles. To plan a collision free path in the workspace, it would need to plan the motion of every point of its shaping according its degree of freedom. The motion of robot between obstacles is represented by a path in configuration space. It is an imaginary concept.

Motion planning is aimed at enabling robots with capabilities of automatically deciding and executing a sequence motion in order to achieve a task without collision with other objects in a given environment. Motion planning in a robot workspace for robotic assembly depends on sequence of parts or the order they are arranged to produce a robotic assembly product obeying all the constraints and instability of base assembly movement. If the number of parts increases the sequencing becomes difficult and hence the path planning. As multiple no. of paths are possible, the path is considered to be optimal when it minimizes the travelling time while satisfying the process constraint. For this purpose, it is necessary to select appropriate optimization technique for optimization of paths. Such types of problem can be solved by metaheuristic methods.

The present work utilizes ACO for the generation of optimal motion planning sequence. The present algorithm is based on ant's behavior, pheromone update & pheromone evaporation and is used to enhance the local search. This procedure is applied to a grinder assembly, driver assembly and car alternator assembly. Two robots like adept-one and puma-762 are selected for picking and placing operation of parts in their workspace.

At last the optimized path considering uncertainties and obstacles within the workspace of industrial robots using ACO technique are developed. This technique generated feasible, stable and optimal robotic assembly sequence and then path sequence satisfying the assembly constraints with minimum travel time. The reverse of the output is the optimal assembly sequence with inverse directions. The solution is either optimal or near optimal.

CONTENTS

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Overview

Motion planning is a term used in robotics for the process of detailing a task into discrete motions. It is a process to compute a collision-free path between the initial and final configuration for a rigid or articulated object (the "robot") among obstacles. It is aimed at enabling robots with capabilities of automatically deciding and executing a sequence motion in order to achieve a task without collision with other objects in a given environment.

Typically the obstacles and the mobile objects are modeled using convex polyhedral or the union of convex polyhedral. Given a source position & orientation for mobile object and goal position & orientation, a search is made for a path from source to goal that is collision free and perhaps satisfied additional criteria such as a short path, a path which can be found quickly or a path which does not wander too close to any one of the obstacles. The general path planning problem requires a search in six dimensional spaces since the mobile object can have three translational and three rotational degrees of freedom. But still there are three dimensional search problems which has two translational and one rotational degrees of freedom.

1.2 Basic steps in robot motion planning

1. Determine the configuration parameters of robot in a given configuration space.

2. Represent the robot and objects properly.

3. Select an motion planning approach suitable to the motion planning problem at hand.

4. Select an appropriate search method to find a solution path.

5. Optimized the solution path for a shorter and smoother path.

1.3. Terms related to robot motion planning problem

Workspace: Workspace is a volume of space which the end-effector of the manipulator can reach.

Workspace is also called work volume or work envelope. The size and shape of the workspace depends on the coordinate geometry of the robot arm, and also on the number of degrees of freedom. Some workspaces are quite flat, confined almost entirely to one horizontal plane. Others are cylindrical; still others are spherical. Some workspaces have very complicated shapes.

Collision: A configuration is said to be in collision if any part of the robot overlaps with either another part of robot or with a work space obstacle.

Configuration: A configuration of a part is a set of parameters which uniquely specify the position of every point on the part.

Configuration space: It is the set of all possible configurations. In configuration space the problem of planning the motion of a part through a space of obstacles is transformed into an equivalent , but simpler, problem of planning the motion of a point through a space of enlarged configuration space obstacles.

Free space: The set of configurations that avoids collision with obstacles is called the free space C_{free}. The complement of C_{free} in C is called the obstacle or forbidden region. Often, it is prohibitively difficult to explicitly compute the shape of C_{free}. However, testing whether a given configuration is in C_{free} is efficient. First, forward kinematics determine the position of the robot's geometry, and collision detection tests if the robot's geometry collides with the environment's geometry.

1.4 Applications of path planning

- Industrial robotics where the robot has to pick up different object and place the object in other places by avoiding collisions.
- In the design of IC chips.
- Machining of a part using NC machines which requires plotting of path of one or more cutting surface so as to produce desired part.

1.5 Objective of the present work

The objective of the present work is to generate feasible, stable and optimal robotic assembly sequence satisfying the assembly constraints with minimum assembly cost. The present research aims at evolving an approach for generating a path planning algorithm or programme so that without collision with obstacles the robot can follow a shortest path from target to goal. The broad objective of research work is outlined as follows.

- To generate the paths of tool center point (TCP) of industrial robot for accomplishing the desired activities.
- To select appropriate technique for optimization of paths as multiple paths are possible to achieve the objective.
- To develop the necessary model for optimization of path for industrial robots considering uncertainties and obstacles within the workspace of industrial robots.

1.6 Methodology

Considering the developments that have taken place and the needs of the process, a systematic way for generation of path sequences in an assembly for motion planning for robotic assembly system is proposed to be developed. A computer-based, generic and integrated optimization method for the generation of assembly sequence is developed. Here, soft computing method i.e. Ant Colony Optimization (ACO) technique for the generation of optimal motion planning sequence minimizing the travel time while satisfying the process constraints is developed. Finally, this method has been proposed with a view to achieve optimized motion planning sequence in relation with constraints, stability criteria and economic factor.

1.7 Outline of the thesis

The thesis describing the present research work is divided into 7 chapters. The subject of the topic its contextual relevance and related matter including the objective of the work and methods to be adopted are presented in chapter 1. The review on several diverse stream of literature on different issues of the topic in chapter 2. In chapter 3 the generation of path sequences are explained which is based on the generation of assembly sequence. Chapter 4 presents the path planning sequences produced by robots are developed. Chapter 5 presents the generation of stable path sequence using ant colony optimization method (ACO). Chapter 6 deals with the result and discussion of the problem. Finally, chapter 7 presents the conclusion and future scope of the research work.

1.8 Summary

The problem of robot path generation and its optimization consists of a consists of a number of factors which cannot be modeled in mathematical terms. There may be multiple alternatives for the same product. As the number of parts increase the number of alternative sequences and hence the number of paths also increases. Therefore, use of the conventional methods to get the optimum one is quite troublesome. This chapter presents the prevailing scenario in motion planning of robot in the presence of obstacles current practice. So, an improved technique has been introduced for a better standard and a systematic way for handling the problem.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

Many studies in the last decade describe efforts to find more efficient algorithm for the path generation amidst the obstacles. The generation of path sequences generally leads to ‘combinatorial explosion’ of the number of alternatives to analyze for checking and selecting the best assembly sequence and consequently, to unacceptable computational time. Different methods have been studied to solve this type of problem; the most efficient one are based on the application of metaheuristic methods that aim to reduce the number of sequences drastically. There have been a lot of research work for the generation of suitable and correct assembly sequences which is reflected through large number of literatures. Many researchers developed different techniques for motion planning of robot considering uncertainties and obstacles within the workspace of robots. The relevant literatures are reviewed and discussed in relation to the methodologies and systems of implementing the above components or activities and towards an integrated environment for supporting the present goal set.

2.2 Some important literatures related to the present work

Table 2.1 presents some of the important work carried out on assembly sequence generation methods.

Table 2.1: Important literatures related to assembly sequence generation

sl	Authors	Year	Topic
1	T. Lozano-Perez and Michael A. Wesley	1979	Describes a collision avoidance algorithm for planning a safe path for a polyhedral object moving among known polyhedral objects.

2	De Fazio and Whitney	1987	Develops a graphical method for the generation of liaison sequences from the precedence relationship among the parts.
3	David Hsu, Jean-Claude Latombe, Stephen Sorkin	1989	Presents an efficient algorithm for optimizing the base location of a manipulator in an environment cluttered with obstacles, in order to execute specified tasked as fast as possible.
4	S. Sharma, R.N. Mohapatra, B.B.Biswal, B.B.Choudhury	2009	Utilize an ant colony optimization (ACO) for the generation of robotic assembly sequences.
5	M. Dorigo	1997	Uses the ant colony algorithm in travelling salesman problem
6	M. Shibata and K.Ohnishi	1992	Developed several mathematical programming method to detect collision, the distance calculation and safe path planning .
7	Shigang Yue, Dominik Henrich, W. L. Xu and S. K. Tso	2002	Focused on the problem of point-to-point trajectory planning for flexible redundant robot manipulators (FRM) in joint space.
8	R.S.Jamisola, Jr.&Anthony A. Maciejewski, Rodney G. Roberts	2003	Presented a method that searches for a continuous obstacle-free space between the starting configuration and the desired configuration.
9	Herry Sutanto and Rajeev Sharma	1997	Considered an approach for motion planning that incorporates visual servoing constraints into the computation of the motion plans.
10	L.M. Galantucci, G. Percoco & R. Spina	2004	proposed the implementation of hybrid Fuzzy Logic-Genetic Algorithm (FL-GA) methodology to plan the automatic assembly and disassembly sequence of products.

2.3 Path planning and assembly sequence generation

De Fazio and Whitney [1] used a logical method through a set of questions that resulted in the desired precedence relationship among the parts. The method generally used two types of questions ('Which connection must be established before connection L_i' and 'Which connections cannot be established before connection L') to be asked. The number of questions to be asked was 2L,the method was far less time consuming but certain relations can be voluntarily omitted. The precedence relationships were used for the generation of assembly sequences. Also, the precedence relations didn't take alternative constraints into account, thus omitting a number of interesting assembly sequences. The techniques did not lead to failure if an obscure liaison was omitted or if conservatively too many are included. However, the method may vary reasonably applied to assemblies with parts counts in the teens or even tens.

De Fazio and Whitney [2] described an integrated computer aid that was useful for assembly-line design and for concurrent design of mechanical products. By recognizing that early consideration of assembly sequence was important for producibility, quality control, flexibility, and market responsiveness, they built an integrated set of user-interactive computer programs that generated all feasible assembly sequences for a product and then aided the user in judging their value based on various criteria. The programs used a disassembly analysis for generating sequences and provided on-line visual aids during generation and evaluation. During evaluation, matters such as avoiding difficult assembly states or moves, stability, fixturing, orientation, refixturing and reorientation count, and inclusion of favorable states were considered to highlight desirable or undesirable sequences. The designer edits the set of sequences according to these criteria, leading to an informed sequence choice or to needed design refinement. The interactive programs provided a rapid mean sequence selection, encouraging their use during early design.

Philip Chan [3] introduced the pattern matching system for the generation of automatic assembly sequence(s) considering that the problem of the part assembly relationship represented as a liaison could be solved in the same way as the travelling salesman problem, and developed a method of reducing the number of questions using the above mentioned method.

Luiz S. Homem de Mello, Arthur C. Sanderson [4] presented an algorithm employed a relational model of assemblies that included a representation of the attachments that bound one part to another. The problem of generating the assembly sequences was transformed into the problem of generating disassembly sequences in which the disassembly tasks were the inverse of feasible assembly tasks. This transformation lead to a decomposition approach in which the problem of disassembling one assembly was decomposed into distinct subproblems, each being to disassemble one subassembly. They assumed that exactly two parts or subassemblies were joined at each time, and that whenever parts were joined forming a subassembly, all contacts between the parts in that subassembly was established. Again they assumed that the feasibility of joining two subassemblies was independent of how those subassemblies were built. The algorithm returns the AND/OR graph representation of assembly sequences. The correctness of the algorithm was based on the assumption that it is always possible to decide correctly whether two subassemblies could be joined, based on geometrical and physical criteria. This paper presented an approach to compute this decision. An experimental implementation for the class of products made up of polyhedral and cylindrical parts having planar or cylindrical contacts among themselves was described.

Christian Mascle, Toni Jabbow & Roland Maramam [5] represented a assembly features. At each stage modeling the product of the assembly process, description of part's faces, assessing accessibility, and modeling technological information made the series of steps that distinguished this model from the others. Such a representation also greatly contributed to the designing of a system that included the various stages of the product elaboration. In this paper three levels of features pertaining to faces, parts and subassemblies were generated to reach the goal.

2.4 Soft computing techniques for optimization

S. Sharma, R.N. Mohapatra, B.B.Biswal, B.B.Choudhury [6] utilized an ant colony optimization (ACO) for the generation of robotic assembly sequences. The method related the assembly cost to an energy function associated with the assembly sequence. The energy function was iteratively minimized to generate an assembly sequence with a minimum assembly cost. There were example problems show the effectiveness of the method. This modified method generated was feasible, stable and optimal robotic assembly sequence satisfying the assembly constraints with minimum assembly cost.

J.F. Wang · J.H. Liu · Y.F. Zhong [7] discussed an ant colony algorithm-based approach for assembly sequence generation and optimization of mechanical products. The approach generated different amount of ants cooperating to find optimal solutions with the least reorientations for diverse assemblies, during assembly processes. Based on assembly by disassembly philosophy, a candidate list composed by feasible and reasonable disassembly operations that were derived from disassembly matrix to guide the sequences construction in the solution space expressed implicitly, and that guaranteed the geometric feasibility of sequences.The state-transition rule and local- and global-updating rules were also defined to ensure acquiring of the optimal solutions.

Dorigo [8-9] introduced ant colony system which was a definition of a new computational paradigm. He proposed it as a viable new approach to stochastic combinatorial optimization. The main characteristics of this model were positive feedback, distributed computation and the use of a constructive greedy heuristic. Positive feedback accounted for rapid discovery of good solutions, distributed computation avoided premature convergence, and the greedy heurisic helped to find acceptable solutions in the early stages of the search process. This method was a distributed algorithm that was applied to the travelling salesman problem. In the ant colony system, a set of co-operating agents called ants cooperate to find good solutions to TSP's. Ants co-operate an indirect form of communication mediated by a pheromone they deposited on the edges of the TSP graph while building solutions.

H. Fujimoto, M. F. Sebaaly [10] introduced a different approach in assembly planning to find the best or optimal sequence to assemble a product, starting from its design data by applying a modified genetic algorithm (GA). A ‘‘best’’ solution was generated without searching the complete candidate space, while search was performed on a sequence population basis. The GA was modified to cope with sequence nonlinearity and constraints.

2.5 Motion planning of robot with obstacle avoidance

T. Lozano-Perez and Michael A. Wesley [11] described a collision avoidance algorithm for planning a safe path for a polyhedral object moving among known polyhedral objects. The algorithm transformed the obstacles so that they represent the locus of forbidden positions for an arbitrary reference point on the moving object. A trajectory of this reference point which avoided all forbidden regions was free of collisions. Trajectories were found by searching a network which indicates, for each vertex in the transformed obstacles which other vertices could be reached safely.

Jing Xiao and Richard A. Volz [12] introduced a replanning approach based on the knowledge of contacts among assembly parts. It consists of Patch planning to resolve the case when a commanded robot motion prematurely stops at a contact other than those planned, and motion strategy planning, to regulate robot motions in order to guarantee the eventual success of a task. A task independent strategy for patch-plan generation based upon concepts of contact planes and abstract obstacles was developed. They developed a unified, systematic method to enable automatic robot assembly.

Sunil K Singh [13] proposed a technique to facilitate decision making. The control was designed using the theory of uncertain dynamical systems and variable structure control to ensure asymptotic convergence which guaranteed uniform ultimate boundness. In this paper the author related this to the controller structure, the maximum available control and the magnitude of the available uncertainty. That information was then used in developing an on-line monitoring frame work for the manipulator. The establishment enabled the monitor to plan, predict and modify the trajectories using nominal linear model and appropriate compensation.

M. Shibata and K.Ohnishi [14] developed several mathematical programming method to detect collision, the distance calculation and safe path planning .These programming includes the application of linear programming, multiple goal programming and the quadratic programming problem. As the workspace in which the restrictive are defined can have an arbitrary number of dimension, it is useful to detect the collisions, to calculate the distance and to plan the trajectories, for any kind of robots.

Kuo-chiang Shao and Kuu.Y. Young [15] proposed to utilize the geometry of the given robot to generate the geometric constraints in the robot workspace. Geometric expressions were then derived o describe the relationship about the planned path and robot workspace. Finally , by applying the developed modification strategies based on different task requirements, feasible paths could be obtained by modifying the infeasible portions of the paths. Here PUMA 560 robot manipulator was selected as a case study due to its complexity and practical application. This proposed scheme was at a better position to take advantage of geometrical properties of the obstacles compared with the inverse kinematics approach.

A.K.C**.** Wong, R.V. Mayorga, L. Rong and X. Liang [16] presented a vision based on-line system for the robust trajectory planning of robot manipulators. It used a 3D vision system to determine the relative position of the objects to be engaged and the obstacle to avoid and a novel obstacle avoidance procedure for manipulator motion planning. These intensity images were acquired by a CCD camera mounted on the robot and the salient features were grouped. Once these 3D poses are determined, an on-line procedure, based on redundancy resolution, was used to achieve obstacle avoidance. The approach utilized a null space vector to set properly the robot configuration, and a potential field method to guide the endeffector. By pseudoinverse perturbation it prevented singular configurations and local minima. The feasibility and effectiveness of the system was demonstrated by an experiment with online engagement and transportation of objects posed inside an aluminium frame.

Herry Sutanto and Rajeev Sharma [17] considered an approach for motion planning that incorporates visual servoing constraints into the computation of the motion plans. It also extends the notion of configuration space to include the corresponding sensor values. Again they proposed a hierarchical representation of the high dimensional planning space involved, and a multi-strategic heuristics search. They applied it practically for several robot manipulators with up to 6-DOF and under various sensing constraints.

David Hsu, Jean-Claude Latombe, Stephen Sorkin [18] presented an efficient algorithm for optimizing the base location of a manipulator in an environment cluttered with obstacles, in order to execute specified tasked as fast as possible. The algorithm used randomized motion planning techniques and exploits geometric "coherence" in configuration space to achieve fast computation.

Rajeev Sharma, Steven M. LaVelle, Seth A.[19] Hutchinson proposed a stochastic representation of the assembly process that improves the performance in the uncertain assembly environment by optimizing an appropriate criterion in the expected sense. The use of the stochastic assembly process provided a flexible way of capturing the time-varying element of assembly operation at different levels.

Adam W. Divelbiss and John T. Wen [20] presented an algorithm for finding a kinematically feasible path for a nonholonomic system in the presence of obstacles. Here they considered the path planning problem without obstacles by transforming it into a nonlinear least squares problem in an augmented space which was then iteratively solved. They considered obstacle avoidance as inequality constraints and exterior penalty functions were used to convert the inequality constraints into equality constraints. Then the same nonlinear least squares approach was applied. This approach was used for solving some challenging problems, including a tractor-trailer and a tractor with a steerable trailer backing in a loading dock.

Shigang Yue, Dominik Henrich, W. L. Xu and S. K. Tso [21] focused on the problem of point-to-point trajectory planning for flexible redundant robot manipulators (FRM) in joint space. Compared with irredundant flexible manipulators, a FRM possessed additional possibilities during point-to-point trajectory planning due to its kinematics redundancy. They presented a trajectory planning method for FRMs to minimize vibration and/or executing time of a point-to-point motion based on Genetic Algorithms (GAs). Kinematics redundancy is integrated into the presented method as planning variables. They used quadrinomial and quintic polynomial to describe the segments that connect the initial, intermediate, and final points in joint space. They formulated trajectory planning of FRM as a problem of optimization with constraints and a planar FRM with three flexible links used in simulation.

E. J. Solteiro Pires, J. A. Tenreiro Machado and P. B. de Moura Oliveira [22] addressed the fractional-order dynamics during the evolution of a Genetic Algorithm (GA) for generating a robot manipulator trajectory. Here the objective was to minimize the trajectory space/time ripple without exceeding the torque requirements. In order to investigate the phenomena involved in the GA population evolution, the mutation is exposed to excitation perturbations and the corresponding fitness variations were evaluated and the input/output signals were studied revealing a fractional order dynamic evolution.

Jun Miura and Yoshiaki Shirai [23] described a method to model the motion uncertainty of moving obstacles and applied to mobile robot motion planning. This method considered three sources of motion uncertainty: path ambiguity, velocity uncertainty and observation uncertainty. They represented the model by a probabilistic distribution over possible position on the path of a moving obstacle. Using this model, the best robot motion was selected which minimized the expected time of reaching the destination considering the distribution of the uncertainty.

Bahaa Ibraheem Kazem , Ali Ibrahim Mahdi and Ali Talib Oudah [24] proposed genetic algorithm (GA) to optimize the point-to-point trajectory planning for a 3-link (redundant) robot arm. The objective function for the proposed GA was to minimize the traveling time and space, while not exceeding a maximum pre-defined torque, without collision with any obstacle in the robot workspace. Quadrinomial and quintic polynomials were used to describe the segments that connect initial, intermediate, and final point at joint-space. Direct kinematics has been used for avoiding the singular configurations of the robot arm.

C.S.Zhao, M. Farooq, M.M.Bayoumi [25] investigated the problem of representing the kinematic motion constraints imposed on the robot arm due to the presence of obstacles. Here kinematic motion constraints caused by any types of obstacles can be analytically and explicitly described by a set of parametric equations. Here Simulations have been carried out for various planar robot arms to verify the validity of the approach.

R.S.Jamisola, Jr.&Anthony A. Maciejewski, Rodney G. Roberts [26] presented a method that searches for a continuous obstacle-free space between the starting Configuration and the desired final end-effector position which is characterized in the joint space by the goal self motion manifold. This method guarantees completion of critical task in the event of a single locked-joint failure in the presence of obstacles.

2.6 Summary

The generations of path sequences are very important for finding the best path sequences and to have an economical and competitive system in place. The above mentioned literatures have been reviewed on generation of possible path sequences in case of industrial assembly based on part design, assembly planning and sequence representation etc. There are many constraints like precedence, geometric and connectivity constraints, cost of the assembly and the least stability criteria are taken into consideration during assembly sequence generation which have studied in many literatures. Again many researchers developed motion planning approach between the starting configuration and goal configuration amidst the obstacles and uncertainties. As multiple paths are possible in the workspace of robot in between the parts of an assembly product, the selection of the best path following all these constraints is a critical factor. To achieve that, the research takes the help of soft computing techniques, i.e. ant colony optimization (ACO). The source of inspiration is taken from the metaheuristic methods to minimize search space explosion in the form of ant behavior. The survey of literatures made in this chapter indicates that a lot of research remains to be done for the generation of optimal sequence.

CHAPTER 3

GENERATION OF PATH SEQUENCES

3.1 Overview

The problem of sequencing the paths has a primary role in the development of collision free path in between the obstacles. Several algorithms have been developed and tested to generate required sequences. The generation of path sequence primarily depends on the assembly sequences. During assembling of a product, an assembly agent will follow a prescribed order to put components into a fixture to complete the final assembly of the product. This order is known as assembly sequence of the product. Exploring the choices of assembly sequence is difficult for two reasons [7]. First, the number of valid sequences can be large even at a small parts count and can rise staggeringly with increasing parts count. Second, seemingly minor design changes can drastically modify the available choices of assembly sequences. At the same time robotic assembly systems are more qualitative and cost effective. This directly influences the productivity of the process, product quality, and the cost of the production. The product to be economically competitive, it is necessary to generate a proper sequence of assembly which minimizes the assembly cost. The assembly conditions may involve the precedence constraints and the connectivity constraints. The precedence constraint is a set of parts that must be connected before a pair of parts are mated. The connectivity constraints, is the connective relationships between the parts. It states that, a part to be assembled onto an in-process sub assembly must have at least one real connection with some part belonging to the in-process sub assembly. The details regarding these aspects are presented in the following sections.

Feasible assembly sequence: Assembly sequences that satisfy the assembly constraints are called the feasible assembly sequences. The feasible assembly sequences do not always guarantee the parts to fix onto an in-process subassembly; parts may be loosely connected, and may come apart during handling.

Stable assembly sequence: The assembly sequences that keep the stability of in-process subassembly movement are called stable sequences, by which the parts can be successfully assembled to form an end product.

A product is considered to be suitable for robotic assembly when the following conditions are satisfied.

- All the individual components are rigid
- Assembly operation can be performed in all mutually perpendicular directions in space excepting +Z direction
- Each part can be assembled by simple insertion or screwing

3.2 Sequence Definitions and Relations: Here, S_{RAN} is a part sequence containing the N parts of a product in a random order. But it might not be feasible. Here,$\{S_{RAN}\}$ is the set of all possible combinations between the N initial parts and S_{AR} is the modified S_{RAN} in order to satisfy the assembly rules existing between the initial parts. It might still violate the assembly constraints. $\{S_{AR}\}$ is the set of all S_{AR} sequences. S_{FE} is the modified S_{AR} in order to meet the assembly constraints while still satisfying the assembly rules. The set of all S_{FE} sequences, $\{S_{FE}\}$, is the search space for the best or optimal sequence. S_{OPT} is the best sequence(s) among all S_{FE}. Thus

$$\{S_{OPT}\} \subseteq \{S_{FE}\} \subseteq \{S_{AR}\} \subseteq \{S_{RAN}\}$$

The schematic diagram of sequence sets is shown in fig. 3.1.

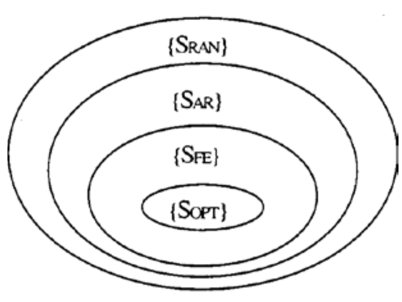


Figure 3.1 Schematic diagram of sequence sets [10]

3.3 Different methods for assembly sequence generation: There are four methods are selected for the application in the generation of assembly sequences**.**

3.3.1 Constraint method: An assembly is called logically infeasible if the pair of subassemblies joined by the tasks is not connected in liaison graph, which is a connected graph G(P, L), where P is a set of all parts, and L is a set of all liaisons of the product. The assembly constraints caused by the geometry of the parts are called G-constraints. The assembly constraint caused by the contact coherence is called C-constraint. C-constraint is determined from the liaison diagram of the product. A part is said to have C-constraint if during the removal of part, its neighboring parts disconnected from the liaison diagram. C-constraints are determined by the connection table or the cut sets

3.3.2 Connectivity graph (CG) method: The graphical representation of interconnections among the parts of a assembly product known as connectivity graph. Each part in the CG is called a node. Three different types of nodes may be found in a typical connectivity graph: (i) sink node (ii) source node (iii) regular node. A sink node is a node with only incoming arrows but no outgoing arrows. A source node is a node with only outgoing, but no incoming arrows. The third type of node is called regular node. A node with both incoming and outgoing arrows represents a regular node, which supports other nodes and is also supported by others.

3.3.3 Liaison method: Liaison sequence analysis is systematic way to generate all the feasible assembly sequences for a product. Liaison is related between parts. Example: Touch, press fit, threaded fit etc. Liaison diagrams show connections between the parts and the liaison sequences are similar to assembly sequences. This method uses precedence relationship among the parts and assembly liaison sequences are generated following some distinct steps

3.3.4 Matrix method: The matrix method is used for the selection of the subassembly sequences of a product. The possible subassemblies are automatically detected by satisfying some mathematical conditions applicable to these matrices. The geometrical model and the technological relationship among the components of a product are represented by means of three matrices as explained below.

A product formed by n elements e_1 ,e_2,……,e_n is represented by the following 3 matrices. Let $A_{k,}$ B_k, C_k be the matrices evaluated along the generic direction k.

Interference Matrix (A_k) : Interference Matrix is that square matrix of order 'n' where a_{ij}=1, if the element e_i interferers with the element e_j during the translation along the direction +k, otherwise a_{ij} =0. As a convention a_{ii} is always zero.

Contact Matrix (B_k) : The contact matrix B_k of a product formed by 'n' elements $e_1, e_2, \ldots, e_n$, is that square matrix of order 'n' where b_{ij}=1, if the element e_i is in contact with the element e_j along the direction +k, otherwise b_{ij} =0. As a convention, b_{ii} is always equal to zero.

Connection matrix (C_k): The connection matrix C_k of a product formed by 'n' elements $e_1, e_2, \ldots, e_n$ in that square matrix of order 'n' where each element of the matrix c_{ij} assumes a numerical code. The code is a function of the kind of connection existing between the elements e_i and e_j along the n^{th} direction 'k'.

3.4 Product modeling for assembly sequence generation

The product modeling is a procedure to explain the assembled state in terms of connective relations between the component parts of given assembly. The connective relations are described in terms of the connective directions and the mating method.

Considering the product consisting *n* parts, the representation of the end product can be made in the following manner.

The product consisting *n* parts is represented in the format A = (P, L),

Where A is a product having parts P = $\{p\alpha \mid \alpha=1, 2 \, . \, . \, . \, . \, . \, n\}$, and interconnected by the liaisons L = $\{l\alpha\beta \mid \alpha, \beta = 1, 2 \, . \, . \, . \, . \, . \, r. \ \alpha \neq \beta\}$ (Cho and Cho).

Here n represents the number of parts of a product and r is the relationship between the connected parts and $(n\text{-}1) \leq r \leq n\,(n\text{-}1)/2$. The liaison $l\alpha\beta$ represents the connective relationship between a pair of parts $p\alpha$ and $p\beta$. The connective relations can be divided into a contact-type and a fit-ty pe connection. The representation of liaison $l\alpha\beta$ is given by

$l\alpha\beta$ = liaison (p_α $C_{\alpha\beta}$, $f_{\alpha\beta}$, p_β),

Where the $C_{\alpha\beta}$ is the contact-type connection matrix and the $f_{\alpha\beta}$ is fit-type connection matrix.

The dimension of each matrix is 2 × 3 elements, and represented by

$$C_{\alpha\beta} = \begin{pmatrix} Cx & Cy & Cz \\ C\bar{x} & C\bar{y} & C\bar{z} \end{pmatrix} \quad \text{and} \quad f_{\alpha\beta} = \begin{pmatrix} fx & fy & fz \\ f\bar{x} & f\bar{y} & f\bar{z} \end{pmatrix}$$

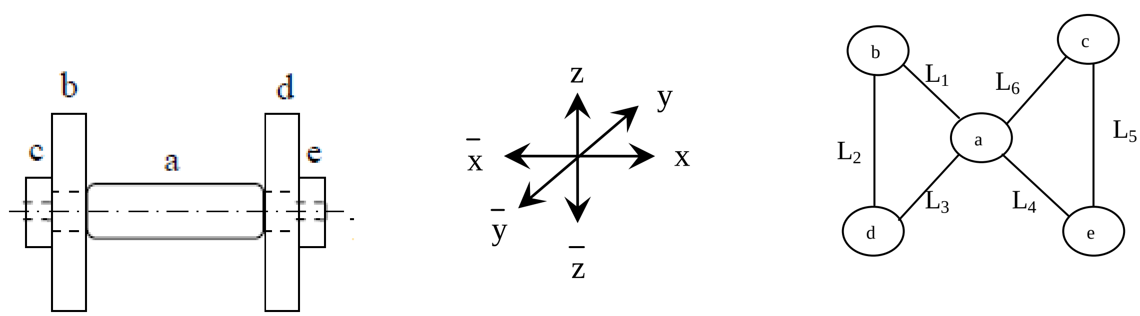
The assembly directions for robotic assembly are taken to be d $\in \{ x, y, \bar{x}, \bar{y}, \bar{z} \}$.The representation of the elements of contact-type and fit-type are:

$$C_d = \begin{cases} 0 : no\ contact\ in\ the\ d\ dir^n\ bet^n\ p_\alpha\ and\ p_\beta \\ rc : real\ contact\ in\ the\ d\ dir^n\ bet^n\ p_\alpha\ and\ p_\beta \\ vc : virtual\ conact\ in\ the\ d\ dir^n\ bet^n\ p_\alpha\ and\ p_\beta \end{cases} \qquad \text{and}$$

$$f_d = \begin{cases} 0 : no\ fit\ in\ the\ d\ dir^n\ bet^n\ p_\alpha\ and\ p_\beta \\ sw : screwing\ in\ the\ d\ dir^n\ bet^n\ p_\alpha\ and\ p_\beta \\ rf : round\ peg\ in\ hole\ fit\ in\ the\ d\ dir^n\ bet^n\ p_\alpha\ and\ p_\beta \\ mp : multiple\ round\ peg\ in\ hole\ fit\ in\ the\ d\ dir^n\ bet^n\ p_\alpha\ and\ p_\beta \end{cases}$$

Each element of f_d can also be represented as round-peg fir (*rf*), a polygon fit (*pf*), a tight fit (*tf*), a caulking (ca), a riveting (ri), a multi-peg-fit (mp), a virtual fit (vf) or no fit (0).

Example Problem 1: The first product (product-1) considered here from [6] is the grinder assembly for the determination of the assembly sequence. Figure 3.2 (a) shows the grinder assembly of the product. Figure 3.2 (b) shows the directions for assembly and Figure 3.2 (c) shows the liaison diagram of the individual component of the product. The table 3.1 shows the part description of the assembly product.



(a) (b) (c)

Figure 3.2 (a) A simple example of a product (Grinder assembly), Figure 3.2(b) Directions for assembly. Fig 3.2 (c) Liaison graph model of grinder.

As per the codes of the model/parts, the liaisons of the assembly components are shown as follows:

$$l_{\alpha\beta} = liaison\left(a, \begin{pmatrix} 0 & rc & rc \\ rc & rc & rc \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ rf & 0 & 0 \end{pmatrix}, b \right)$$

$$l_{ac} = liaison\left(a, \begin{pmatrix} 0 & 0 & 0 \\ vc & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ sw & 0 & 0 \end{pmatrix}, c \right)$$

$$l_{ad} = liaison\left(a, \begin{pmatrix} rc & rc & rc \\ 0 & rc & rc \end{pmatrix}, \begin{pmatrix} rf & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, d \right)$$

$$l_{ae} = liaison\left(a, \begin{pmatrix} vc & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} sw & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, e \right)$$

$$l_{bc} = liaison\left(b, \begin{pmatrix} 0 & 0 & 0 \\ rc & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, c \right)$$

$$l_{de} = liaison\left(d, \begin{pmatrix} rc & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, e \right)$$

Example problem 2: The second product (product-2) considered here [7] is the driver assembly for the determination of the assembly sequence. It consists of 16 components where the screws fastening the same two components are grouped as one.

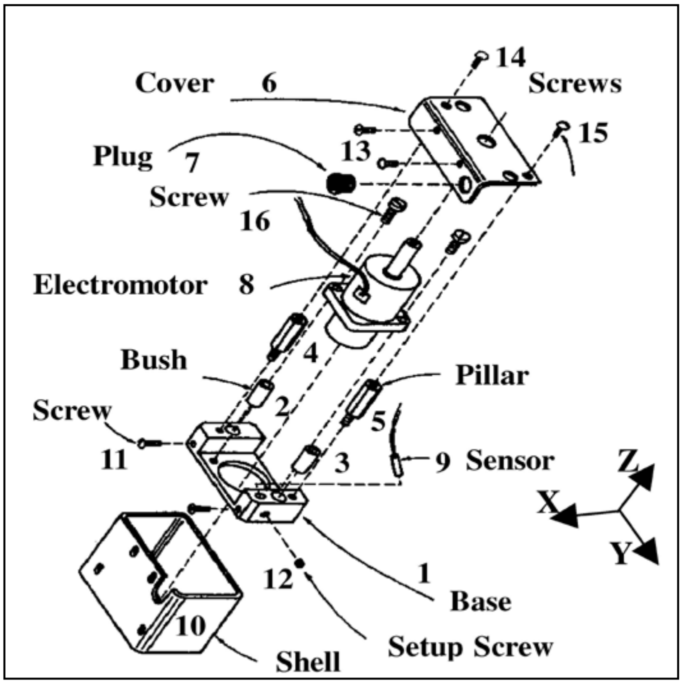


Figure 3.3(a) The driver assembly

Figure 3.3(a) shows the complete driver assembly and figure 3.3(b) shows the liaison diagram of the driver assembly.

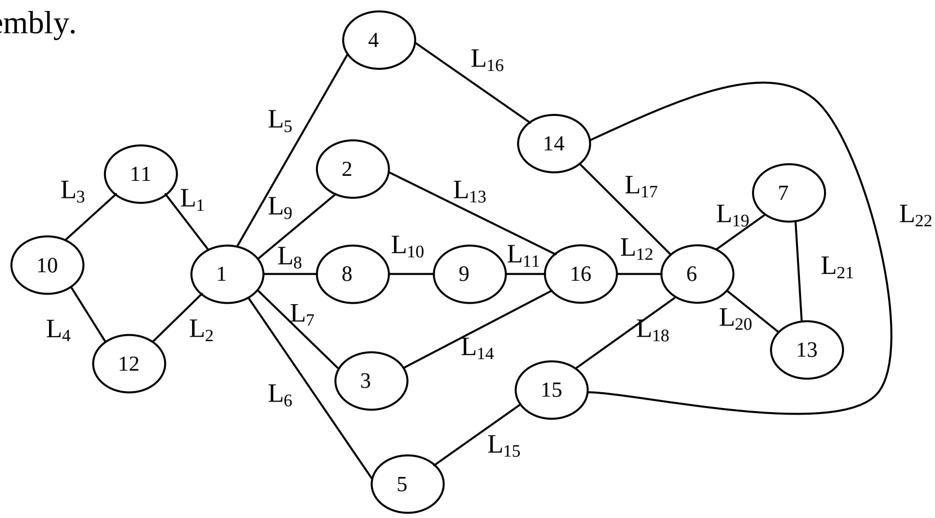


Figure 3.3(b) The liaison diagram of the driver assembly

Example problem 3: The third product (product-3) considered here [10] is the car alternator assembly for the determination of the assembly sequence. Figure 3.4(a) shows the car alternator assembly. Figure 3.4(b) shows the liaison graph of car alternator assembly.

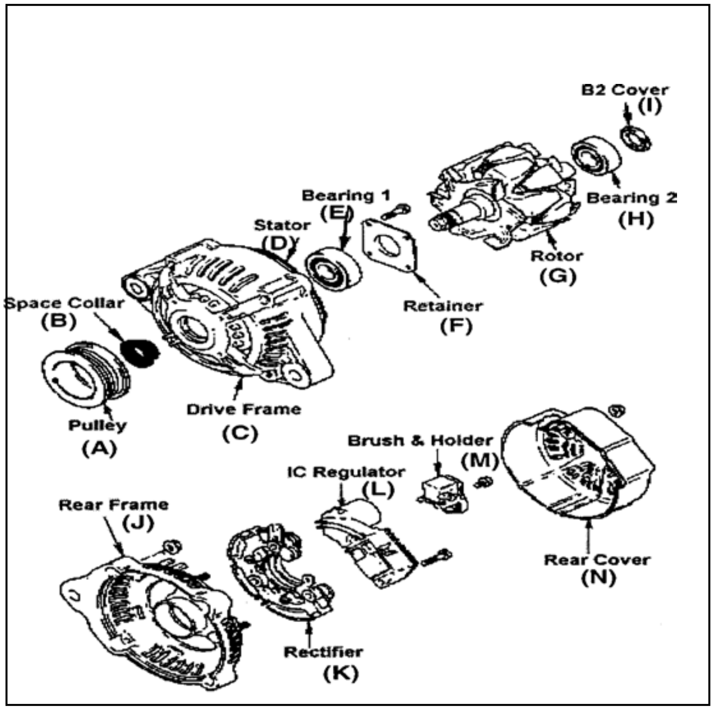


Figure 3.4(a) The car alternator assembly

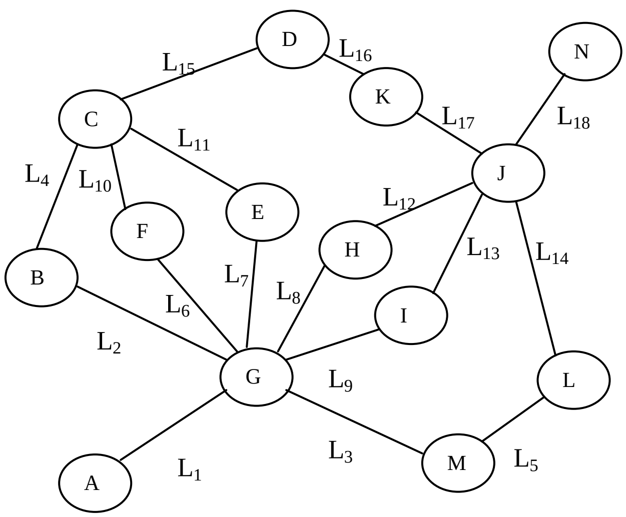


Figure 3.4(b) Liaison diagram of car alternator assembly

3.5 Selection of robot for pick and place operation

For placing the different parts of assembly at their particular location following robots are selected .

i. SCARA robot: It known as selective compliance assembly robot arm. It is also known as articulated robot. These robots are suitable for assembly. These robots are provided with direct drive motors that allow high speeds with acceleration and backlash-free, fast and accurate motions. Example: Adept-1.Figure 3.5(a) shows the SCARA robot.

ii. Revolute robot: This robot resembles the human arm. In revolute robot, all the joints are revolute. It has six degrees of freedom. Three are in X,Y and Z axes. The other three are pitch, yaw and roll. Pitch is when the wrist moves up and down. Yaw is when the hand moves left and right. Roll is when the forearm entirely rotates. PUMA series robots are example of this type robot. Figure 3.5(b) shows the revolute robot.

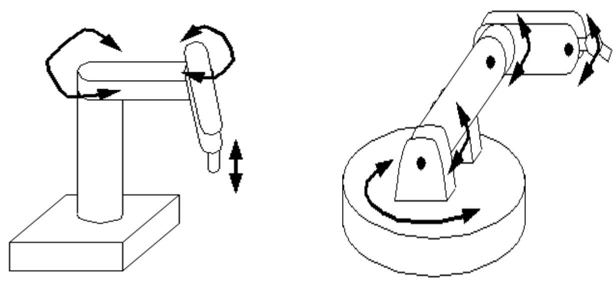


Figure 3.5 (a) Schematic diagram of SCARA Robot,

Figure 3.5 (b) Schematic diagram of Revolute Robot

Table 3.2 shows the comparison of Adept-one and Puma-762 Robots.

Table 3.1: Comparison of Adept-one and Puma-762 Robots

Types of robot	**Adept-one**	**Puma-762**
Payload	9.1 kg	20 kg
Maximum Reach	800 mm	1388 mm
Maximum speed	1100 mm/sec	1000 mm/sec
Degree of freedom	4	6
Configuration	R-R-P	R-R-R
Range of angles	$+150^0$ to -150^0	$\pm 320^0$
Repeatability	x,y = ± 0.02mm, z = ± 0.01mm	± 0.2mm

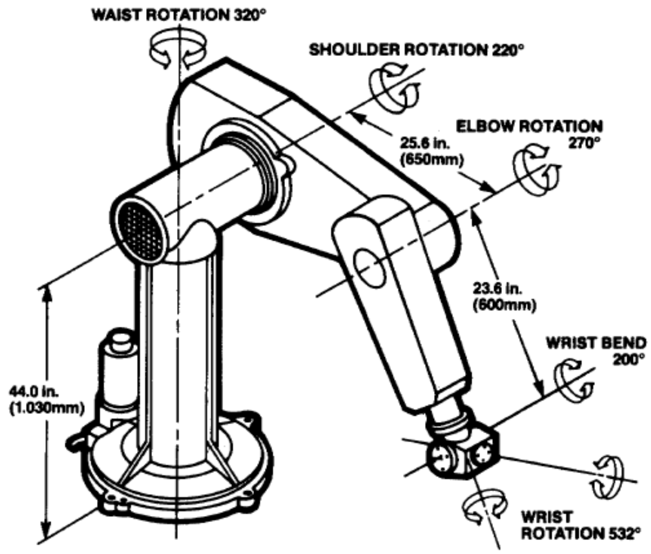


Figure 3.6(a).PUMA-762 robot arm. Degrees of joint rotation and member identification.

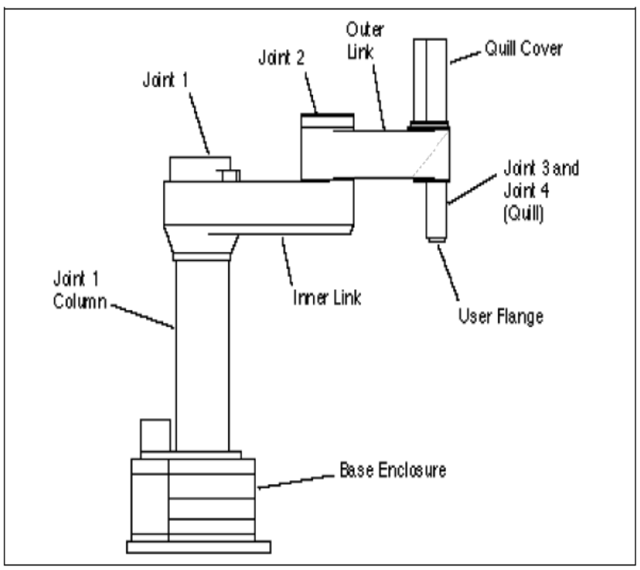


Figure 3.6(b) Adept One Robot Joint Locations

Figure 3.6(a) and 3.6(b) shows Puma-762 and Adept one robots.

Figure 3.7(a).shows Adept One robot working Envelope and Fig 3.7(b) shows Puma-762 robot working Envelope.

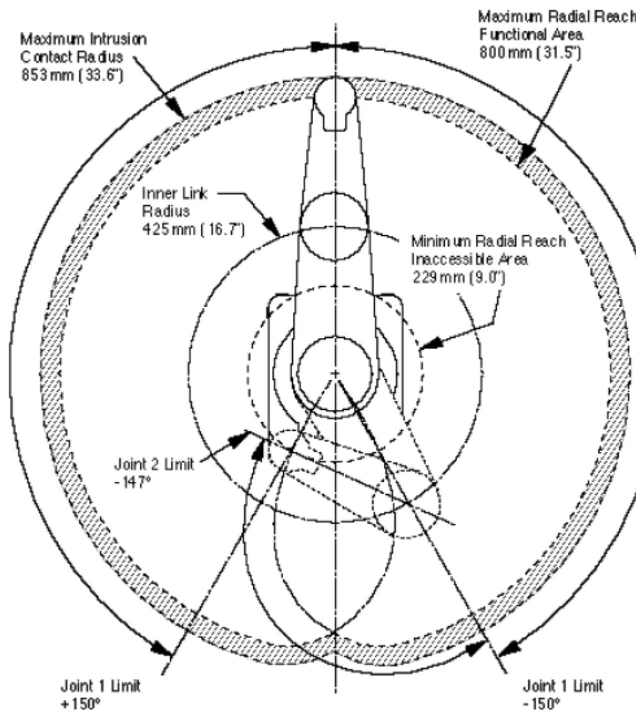


Figure 3.7(a). Adept One Robot Working Envelope

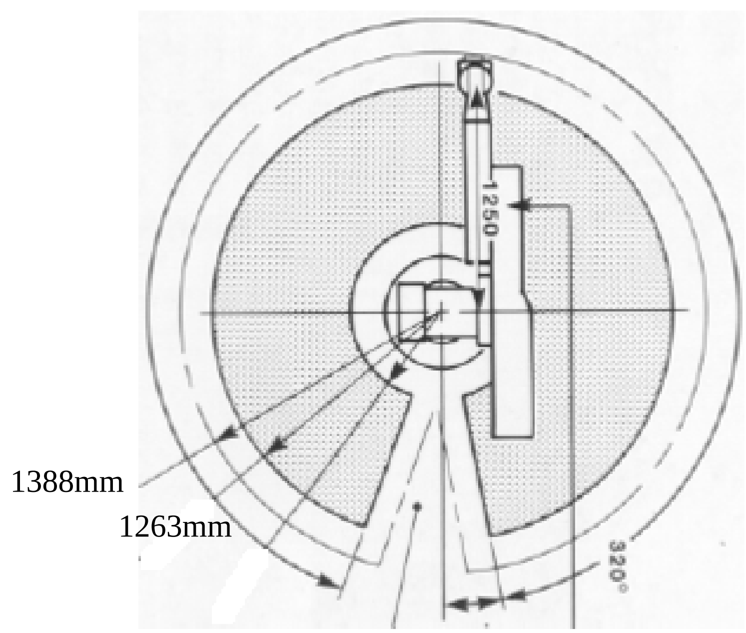


Figure 3.7(b) Puma-762 Robot Working Envelope

3.6.1. TASK DECOMPOSITION FOR ADEPT-ONE OR PUMA-762 ROBOT FOR WORK CELL-1 AND WORK CELL-2

Product-1: *Grinder Assembly*

Table3.2: Task decomposition for grinder assembly for work cell-1 and work cell-2

Sl no.	Part Name	Part ID	Task
1	Shaft	a	Pick-rotate-orient-move-place
2	Blade	b	Pick- rotate-move-place
3	Nut	c	Pick -move-insert-place
4	Blade	d	Pick- rotate-move-place
5	Nut	e	Pick- move-insert-place

Table 3.2 shows the task decomposition for grinder assembly for work cell-1 and work cell-2.

Product-2: Driver assembly

Table3.3: Task decomposition for driver assembly for work cell-1 and work cell-2

Sl no.	Part Name	Part ID	Task	Sl no.	Part Name	Part ID	Task
1	Base	a	Pick- rotate-move-place	9	Sensor	i	Pick- move-insert-place
2	Bush1	b	Pick- move-insert-place	10	Shell	j	Pick- move-insert-place
3	Bush2	c	Pick- move-insert-place	11	Screw	k	Pick- move-insert-place
4	Pillar1	d	Pick- move-insert-place	12	Setup Screw	l	Pick- move-insert-place
5	Pillar2	e	Pick- move-insert-place	13	Screw	m	Pick- move-insert-place
6	Cover	f	Pick-rotate-move-place	14	Screw	n	Pick- move-insert-place
7	Plug	g	Pick- move-rotate-insert-place	15	Screw	o	Pick- move-insert-place
8	Electro-motor	h	Pick-rotate-move-place	16	Screw	p	Pick- move-insert-place

Table3.3 shows the task decomposition for driver assembly for work cell-1 and work cell-2

Product-3: Car alternator assembly

Table3.4: Task decomposition for car alternator assembly for work cell-1 and work cell-2

Sl no.	Part Name	Part ID	Task	Sl no.	Part Name	Part ID	Task
1	Pulley	A	Pick-rotate- move - insert-place	8	Bearing 2	H	Pick- move- insert-place
2	Space collar	B	Pick-rotate- move- insert-place	9	B2 cover	I	Pick- move- insert-place
3	Drive Frame	C	Pick-rotate-move-place	10	Rear frame	J	Pick- move-insert-place
4	Stator	D	Pick- rotate-move- insert-place	11	Rectifier	K	Pick- rotate-move-attach-place
5	Bearing 1	E	Pick-rotate- move- insert-place	12	IC Regulator	L	Pick- rotate-move-attach-place
6	Retainer	F	Pick- rotate-move- insert-place	13	Brush & Holder	M	Pick- move-insert-place
7	Rotor	G	Pick-rotate- move-place	14	Rear cover	N	Pick-rotate-move-place

Table3.4 shows the task decomposition for alternator assembly for work cell-1 and work cell-2

3.6.2 TASK DECOMPOSITION FOR ADEPT-ONE AND PUMA-762 ROBOT FOR WORK CELL-3

Product-1*: Grinder Assembly*

Table 3.5: Task decomposition for grinder assembly for work cell-3

Sl no.	Part Name	Part ID	Task	Task assignment to Robot	Reason
1	Shaft	a	Pick-rotate-orient-move-place	Puma-762	Better suitability

2	Blade	b	Pick- rotate-move-place	Adept-one	Better suitability
3	Nut	c	Pick -rotate-move-insert-place	Adept-one	Better suitability
4	Blade	d	Pick- rotate-move-place	Adept-one	Better suitability
5	Nut	e	Pick-rotate- move-insert-place	Adept-one	Better suitability
6	Sub assembly-1	a-d-e	Pick-rotate-orient-move-place	Puma-762	Better suitability

Table 3.5 shows the task decomposition for grinder assembly for work cell-3

Product-2: Driver assembly

Table 3.6: Task decomposition for driver assembly for work cell-3

Sl no.	Part Name	Part ID	Task	Task assignment to Robot	Reason
1	Base	a	Pick-rotate-orient-move-place	Puma-762	Better suitability
2	Bush1	b	Pick-rotate-move-insert-place	Puma-762	Better suitability
3	Bush2	c	Pick- orient-move-insert-place	Puma-762	Better suitability
4	Pillar1	d	Pick- move-insert-place	Adept-one	Better suitability
5	Pillar2	e	Pick- move-insert-place	Adept-one	Better suitability
6	Cover	f	Pick-rotate-move-place	Adept-one	Better suitability
7	Plug	g	Pick-rotate-move-insert-place	Adept-one	Better suitability

8	Electromotor	h	Pick-rotate-move-insert-place	Adept-one	Better suitability
9	Sensor	i	Pick- move-insert-place	Adept-one	Better suitability
10	Shell	j	Pick- move-insert-place	Adept-one	Better suitability
11	Screw	k	Pick- move-insert-place	Adept-one	Better suitability
12	Setup Screw	l	Pick- move-insert-place	Adept-one	Better suitability
13	Screw	m	Pick- move-insert-place	Adept-one	Better suitability
14	Screw	n	Pick- move-insert-place	Adept-one	Better suitability
15	Screw	o	Pick- move-insert-place	Adept-one	Better suitability
16	Screw	p	Pick- move-insert-place	Adept-one	Better suitability
17	Sub-assembly1	a-h-p-c-e-d-i-b	Pick- rotate-orient-move-place	Puma-762	Better suitability
18	Sub-assembly-2	f-n-o	Pick- rotate-orient-move-place	Puma-762	Better suitability

Table 3.6 shows the task decomposition for driver assembly for work cell-3

Product-3: Car alternator assembly

Table3.7: Task decomposition for optimal sequence of car alternator assembly for work cell-3

Sl no.	Part Name	Part ID	Task	Task assignment to Robot	Reason
1	Pulley	A	Pick-rotate- move-insert-place	Adept-one	Better suitability
2	Space collar	B	Pick- rotate-move -insert-place	Adept-one	Better suitability
3	Drive Frame	C	Pick-rotate-orient-move-place	Puma-762	Better suitability
4	Stator	D	Pick-rotate- move -insert-place	Adept-one	Better suitability
5	Bearing 1	E	Pick-rotate-move-insert-place	Adept-one	Better suitability
6	Retainer	F	Pick-rotate-move-insert-place	Adept-one	Better suitability
7	Rotor	G	Pick-rotate-move-place	Adept-one	Better suitability
8	Bearing 2	H	Pick- move- insert-place	Adept-one	Better suitability
9	B2 cover	I	Pick- move- insert-place	Adept-one	Better suitability
10	Rear frame	J	Pick- rotate-orient-move-insert-place	Puma-762	Better suitability
11	Rectifier	K	Pick- rotate-move-attach-place	Adept-one	Better suitability
12	IC Regulator	L	Pick- rotate-move-attach-place	Adept-one	Better suitability
13	Brush & Holder	M	Pick- move-insert-place	Adept-one	Better suitability
14	Rear cover	N	Pick-rotate-move-place	Adept-one	Better suitability
15	Sub-assembly-2	G-H-I-J-K-L-M-N	Pick-rotate-orient-move-place	Puma-762	Better suitability
16	Sub-assembly-1	C-B-E-F-D	Pick-rotate-orient-move-place	Puma-762	Better suitability

Table3.7 shows the task decomposition for optimal sequence of car alternator assembly for work cell-3

3.7.1 WORKCELL-1 WITH ADEPT-ONE ROBOT:

Product-1: Grinder assembly

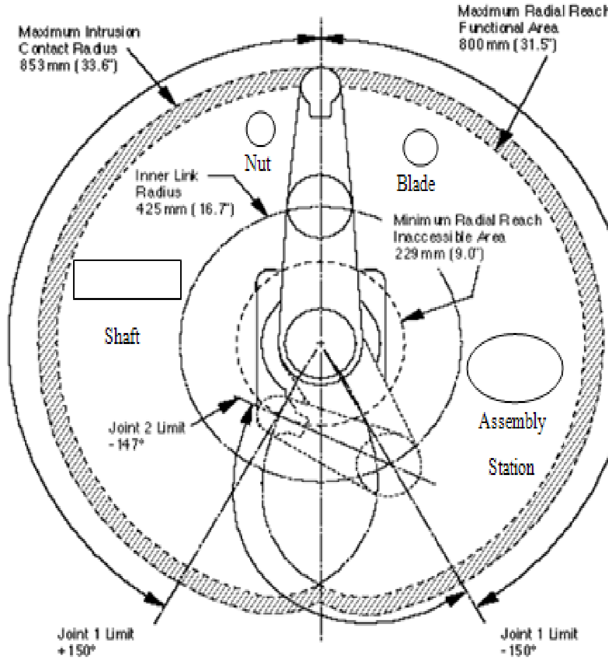


Figure 3.8. Workcell-1 for grinder assembly with Adept-one Robot

Product-2: Driver assembly

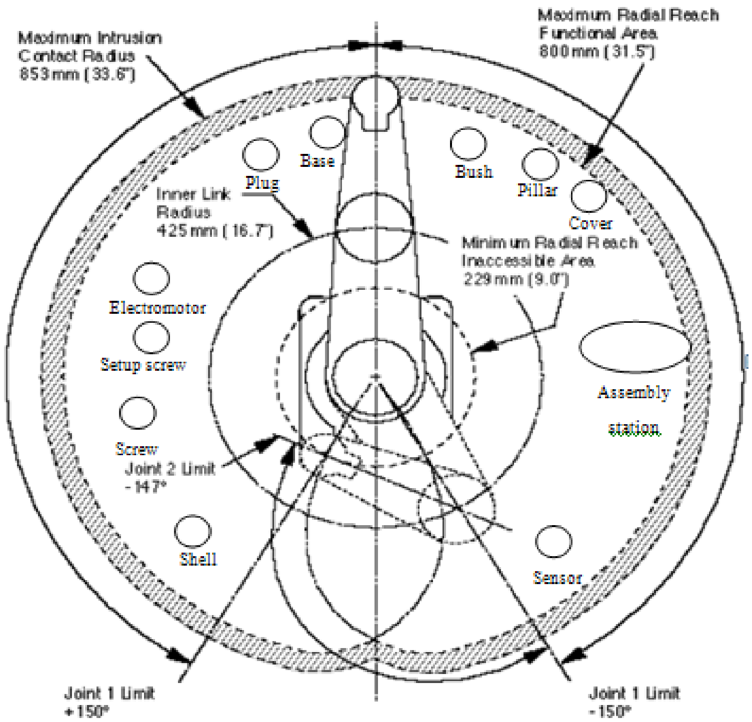


Figure 3.9.Workcell-1 for driver assembly with Adept-one Robot

Product-3:Car alternator assembly

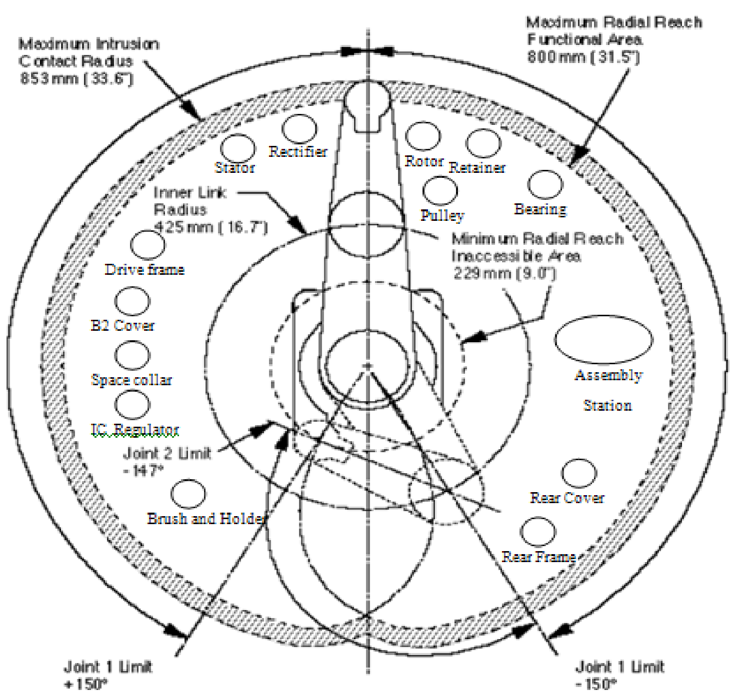


Figure 3.10 Workcell-1 for car alternator assembly with Adept-one Robot

3.7.2 WORKCELL-2 WITH PUMA-762 ROBOT:

Product-1: Grinder assembly

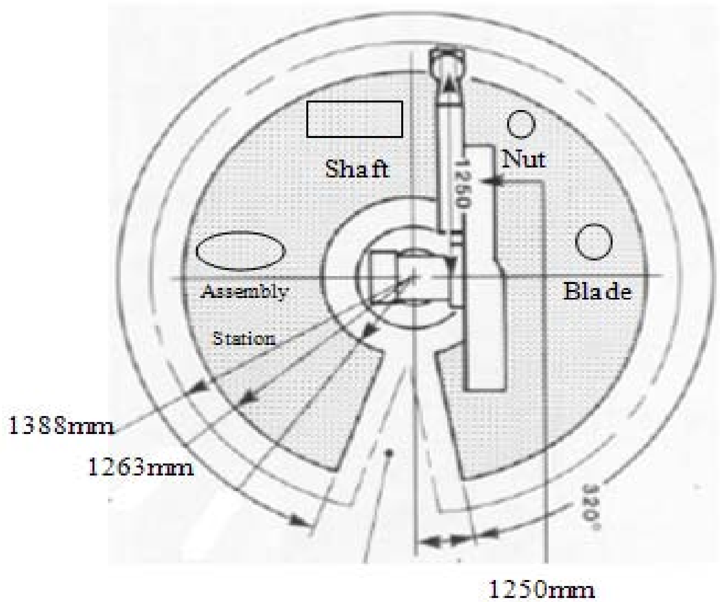


Figure 3.11.Workcell-2 for grinder assembly with Puma-762 Robot

Product-2: Driver assembly

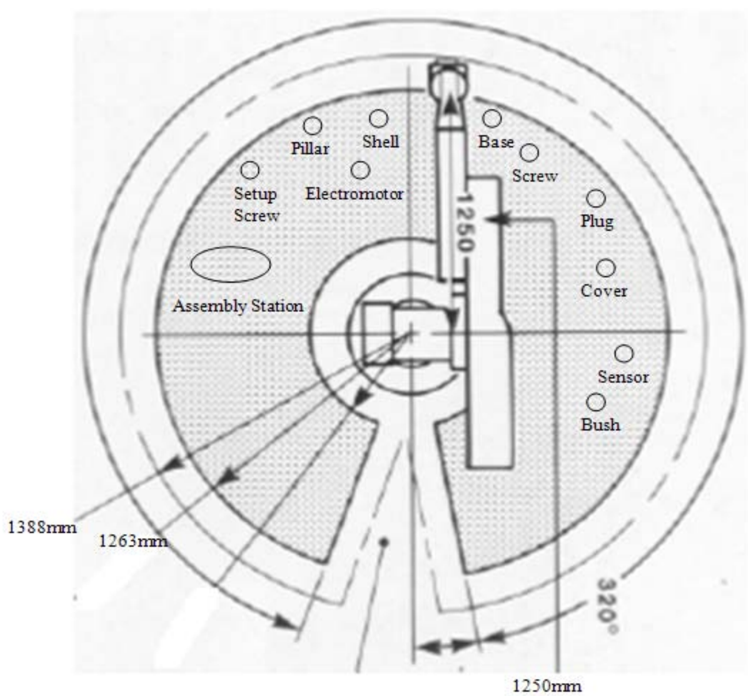


Figure 3.12 Workcell-2 for driver assembly with Puma-762 Robot

Product-3:Car alternator assembly

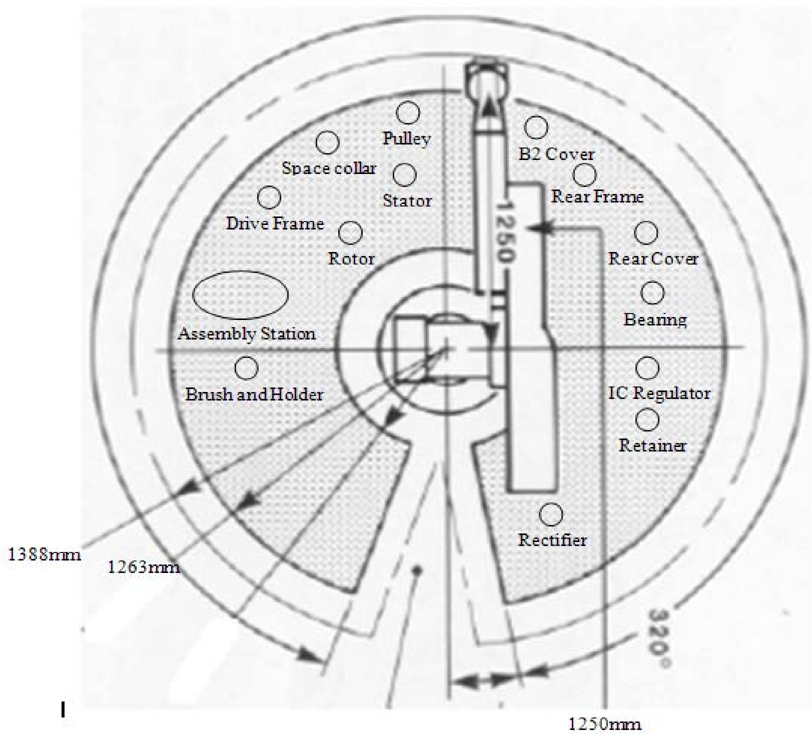


Figure 3.13 Workcell-2 for car alternator assembly with Puma-762 Robot

Figure 3.8, 3.9 and 3.10 shows workcell-1 for grinder assembly,driver assembly and car alternator assembly with Adept-one Robot and Figre 3.11, 3.12 and 3.13 shows workcell-2 for grinder assembly, driver assembly and car alternator assembly with Puma-762 Robot.

3.7.3 WORKCELL-3 WITH ADEPT-ONE AND PUMA-762 ROBOT:

Product-1: Grinder assembly

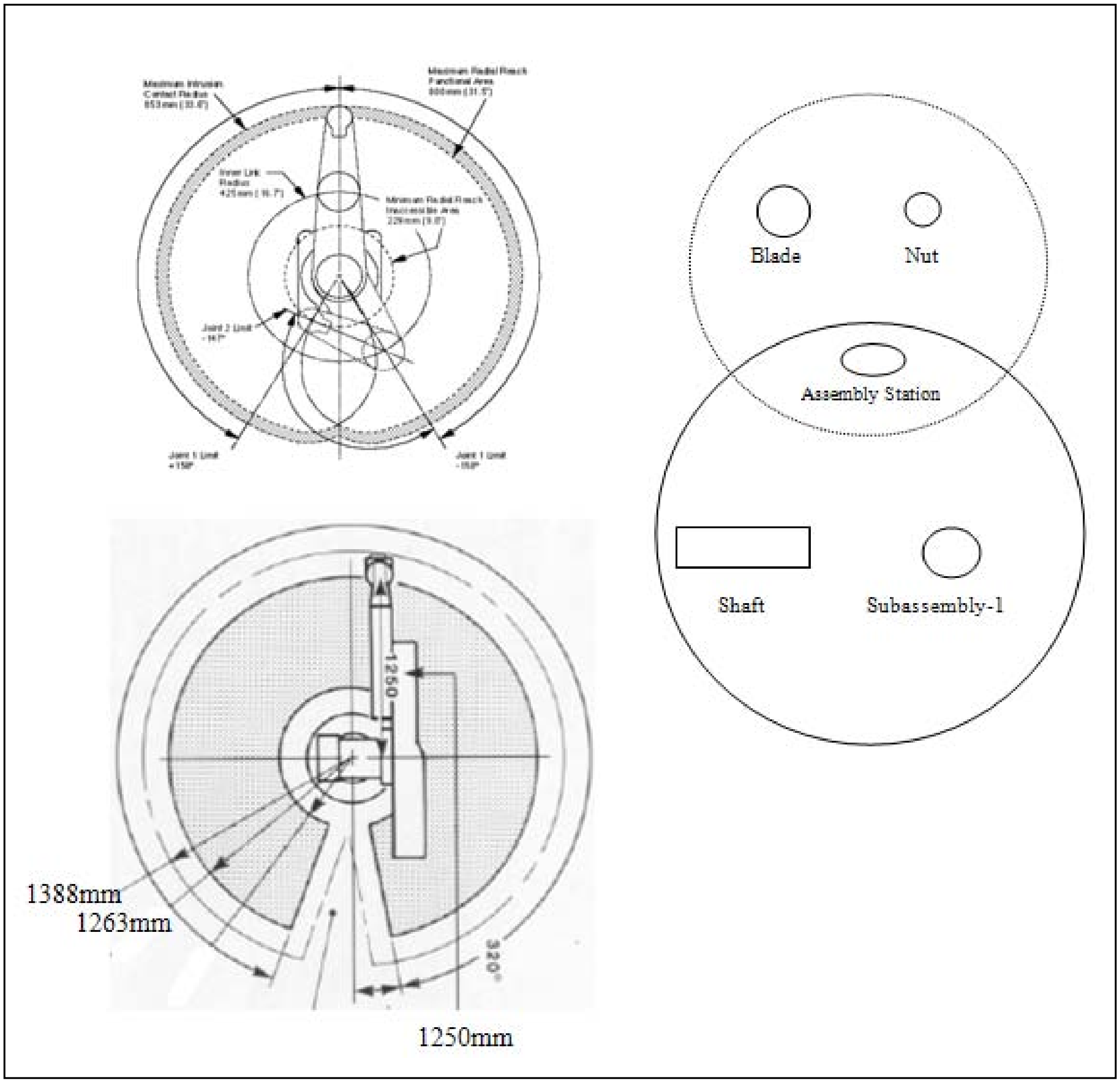


Figure 3.14. Workcell-3 for grinder assembly with Adept-one and Puma-762 Robot

Product-2: Driver assembly

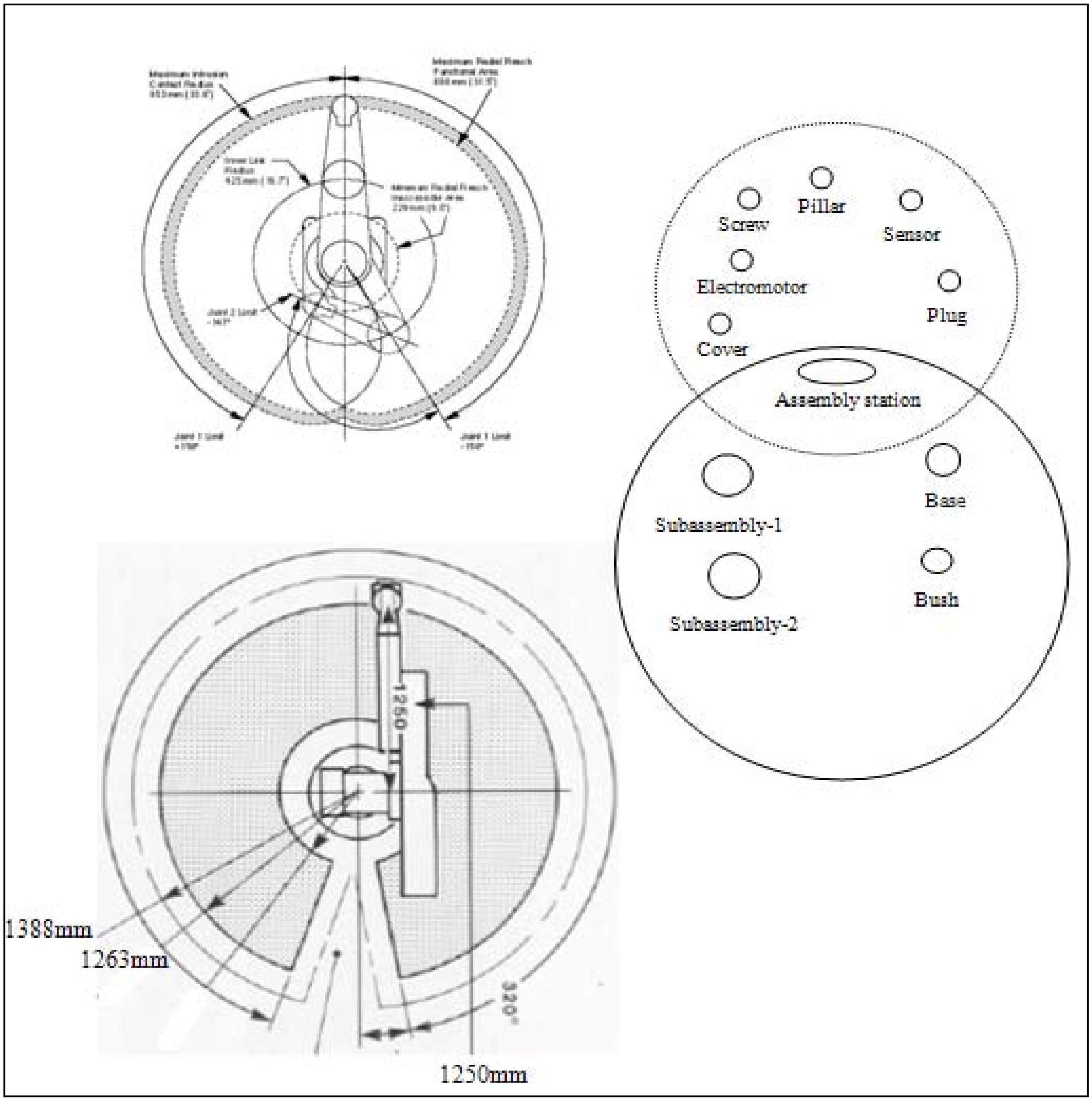


Figure 3.15. Workcell-3 for driver assembly with Adept-one and Puma-762 Robot

Product-3:Car alternator assembly

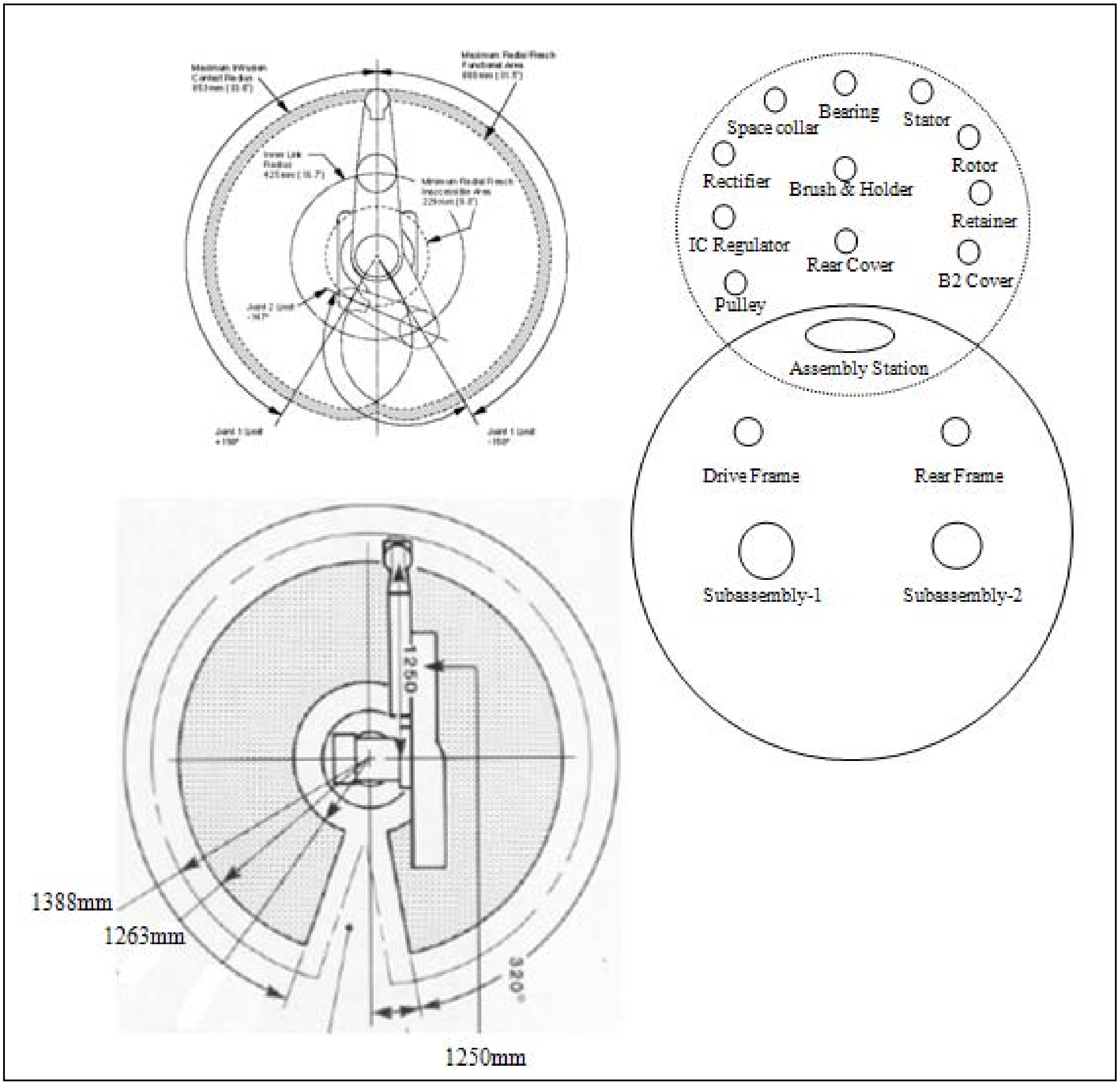


Figure 3.16. Workcell-3 for car alternator assembly with Adept-one and Puma-762 Robot

3.8 Flow Chart For Feasibility Study: Fig 3.18 shows the flowchart for feasibility study between the parts of the assembly. Here PC denotes precedence constraint,GC denotes geometric constraint and CC denotes connectivity constraint.

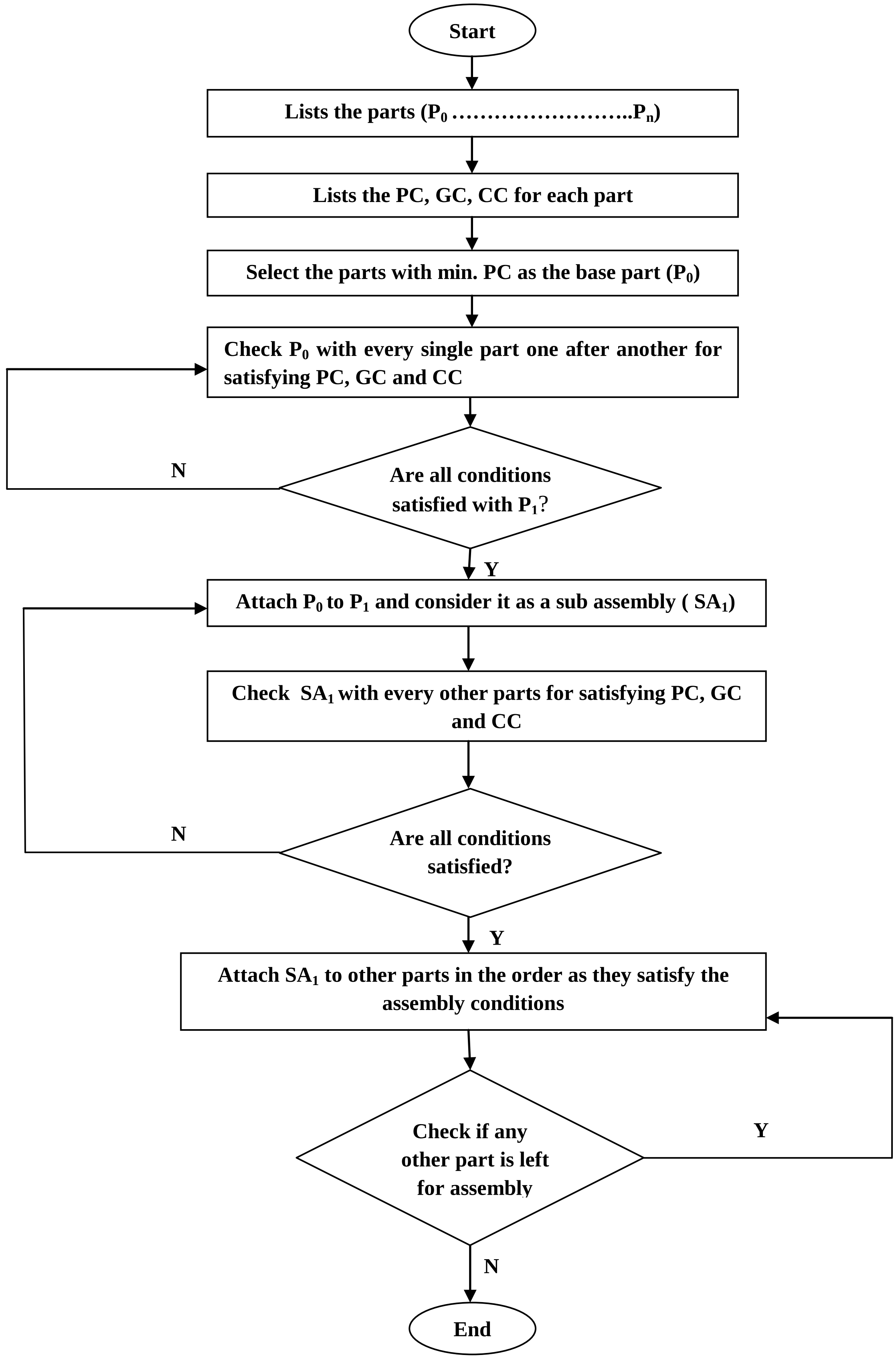


Figure 3.17 shows the flow chart for feasibilty study

3.9 Summary

For generating correct assmbly sequence for robotic assembly a systematic method has been proposed. The topological relationship between parts of the assembly of the product has taken into consideration and it produce the stable sequence. Three products viz, (i)grinder assembly, (ii)driver assembly, (iii) car alternator assembly are taken as example problems. The relation between the parts is shown in the respective liaison diagrams. As there are possibilities of number of assembly sequences the stable sequence is one which yields least number of direction changes. Then two types of robots i.e. Adept-one and Puma-762 are selected for placing the parts of the assembly product in their workspace. Tasks are decomposed to these two robots individually in workcell-1 and workcell-2 respectively, and the combination of two robots in workcell-3. The feasiblity study betrween the parts of the assembly product has been shown in the flowchart. The development of a procedure to cluster parts into subassemblies to obtain a hierarchial model of the assembly and the development of good heuristics to guide the generation of assembly sequences,followed by the motion planning sequences are issues for the future research.

CHAPTER 4

PATH PLANNING

4.1 Overview

Path planning is defined as finding a continuous motion that will take a manipulator from a given initial or source position to final or goal position subjected to the constraint that any point in the motion the manipulator does not collide with any obstacle in its workspace. It is the design of only geometric (kinematical) specifications of the position and orientations of the robot. The path planning module is used to determine a route from one coordinate location to another along a set of waypoints. Example: if you had an image of a maze and you need to determine the best path from where the robot is currently located to where it need to be you would use the path planning module to determine the shortest or best path to the desired location.

4.2 Uncertainty

Task planning is a challenging problem even if our knowledge of the position and orientation of the parts within the workspace is exact [28]. In realty, the variables which represent the part position and orientation will have a nominal value plus an error term which represent uncertainty:

$$v^{exact} = v^{nominal} + \Delta v \quad \ldots\ldots\ldots\ldots\ldots\ldots\text{(i)}$$

$$\|\Delta v\| \leq \Delta v^{max} \quad \ldots\ldots\ldots\ldots\ldots\ldots\text{(ii)}$$

The error bound Δv^{max} represents tolerances in the size of a machined part or it might be associated with the error in a sensor such as an overhead camera used to locate a part. Different

types of uncertainty may arise during motion planning..Those are given below.

- Motion uncertainty
- Missing information
- Active sensing

- Sensor less planning

Here another example_for finding shortest path if there is uncertainty in the position, orientation, size, or shape of the polygon. Fig 4.1 shows finding the shortest path using configuration space and forming the collision free path.

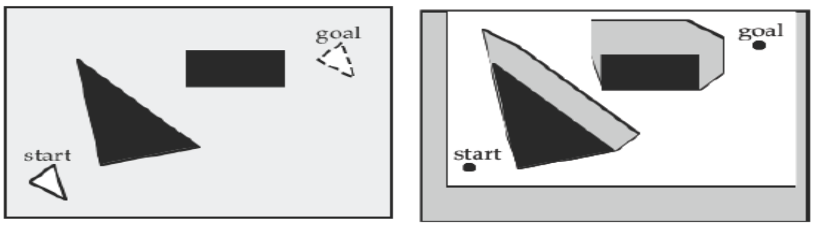


Figure 4.1.Forming the collision free path by configuration space in presence of uncertainties

4.3. Problems on configuration space

Problem 1: Consider the scene shown in fig(9).Suppose A is the mobile part and B_1 and B_2 are fixed obstacles. Sketch the configuration space scene induced by A, using reference point r.

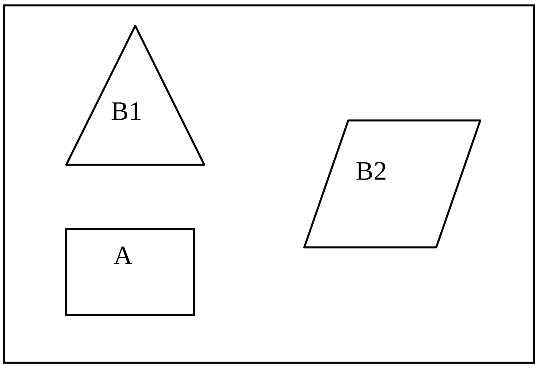


Figure 4.2. A workspace with two obstacles

Solution:

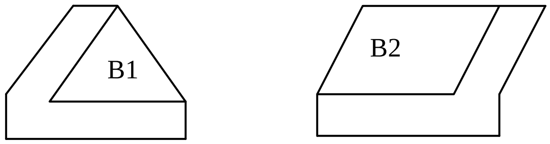


Figure 4.3(a) Generating the configuration space obstacle B1A

Figure 4.3(b) Generating the configuration space obstacle B2A

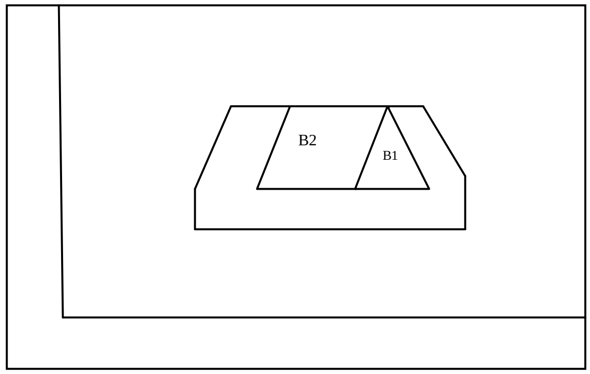


Figure 4.4 Configuration space induced by part A

Problem 2: Repeat problem-1, but with the mobile part rotated π/2. That is sketch the configuration space slice projection associated with a mobile orientation of ø = π/2.

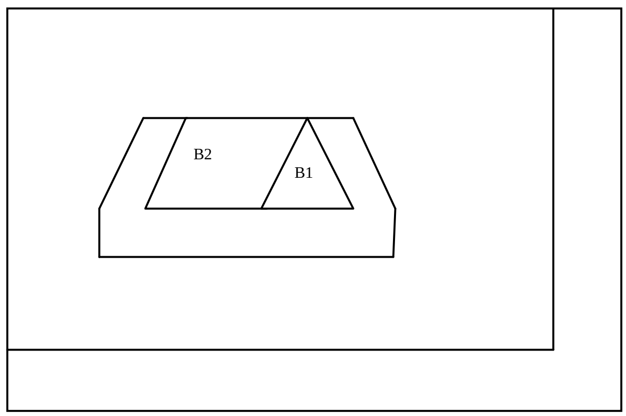


Figure 4.5 Configuration space induced by part A rotated by π/2

4.4. Path Sequences in workspace for both workcell-1 and workcell-2 :

Product-1.Grinder Assembly

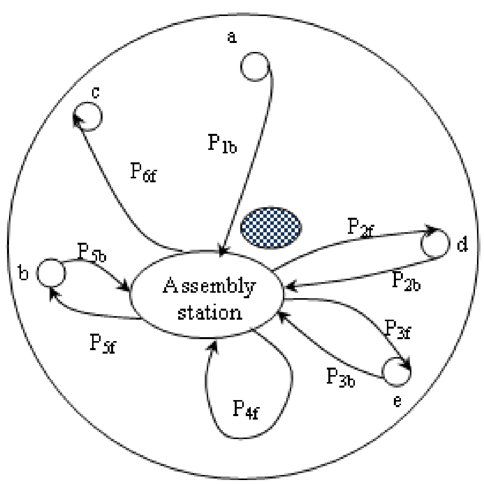


Figure 4.6. Path sequences for grinder assembly for Adept-one or Puma-762 robots

Product-2. Driver Assembly

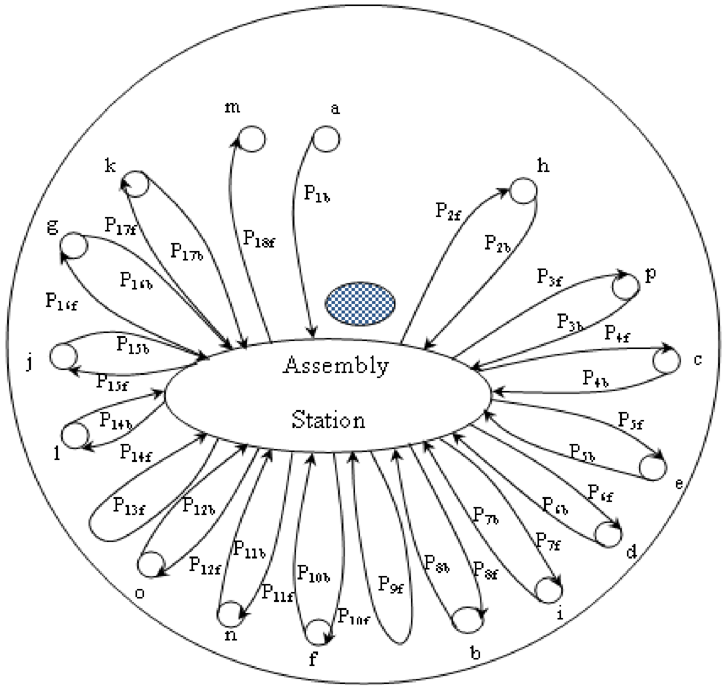


Figure 4.7. Path sequences for driver assembly for Adept-one or Puma-762 robots

Product-3 .Car Alternator Assembly

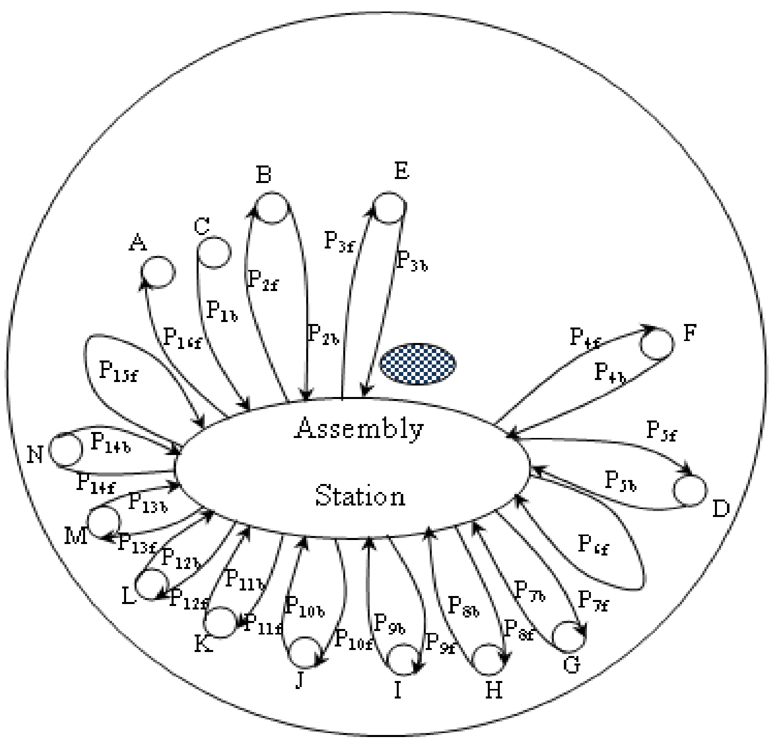


Figure 4.8. Path sequences for car alternator assembly for Adept-one or Puma-762 robots

Figure 4.6,4.7 and 4.8 shows the path sequences for grinder assembly, driver assembly and car alternator assembly for Adept-one or Puma-762 robots

4.5 Path Sequences for workcell-3

Product-1 (Grinder Assembly)

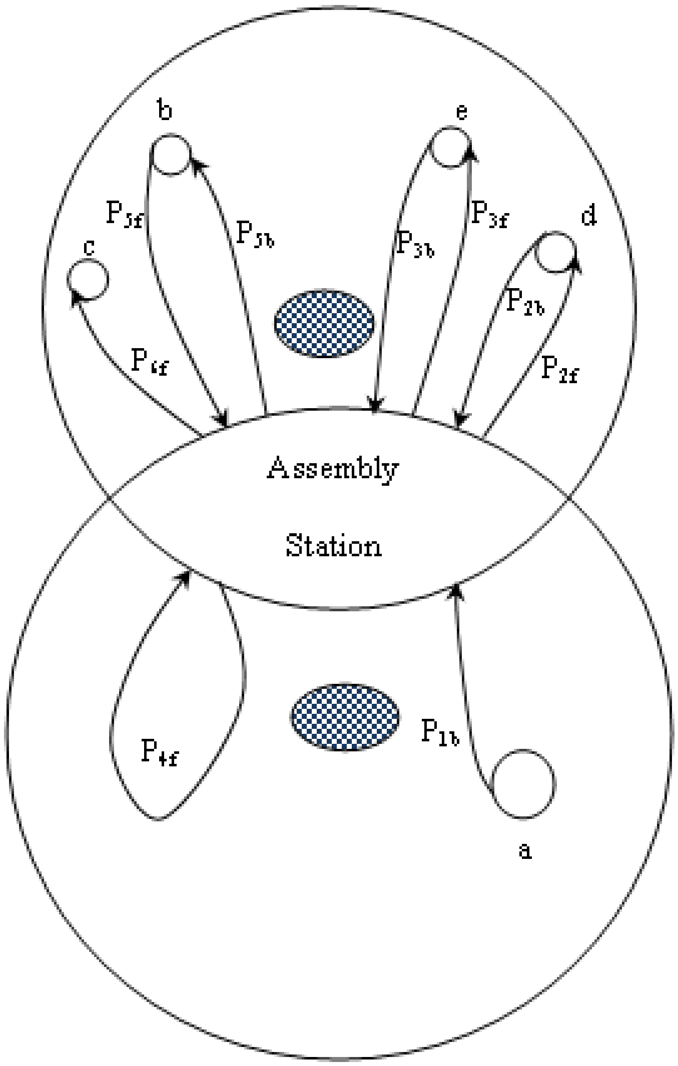


Figure 4.9. Path sequences for grinder assembly for both Adept-one and Puma-762 robots

Product-2 (Driver Assembly) for workcell-3

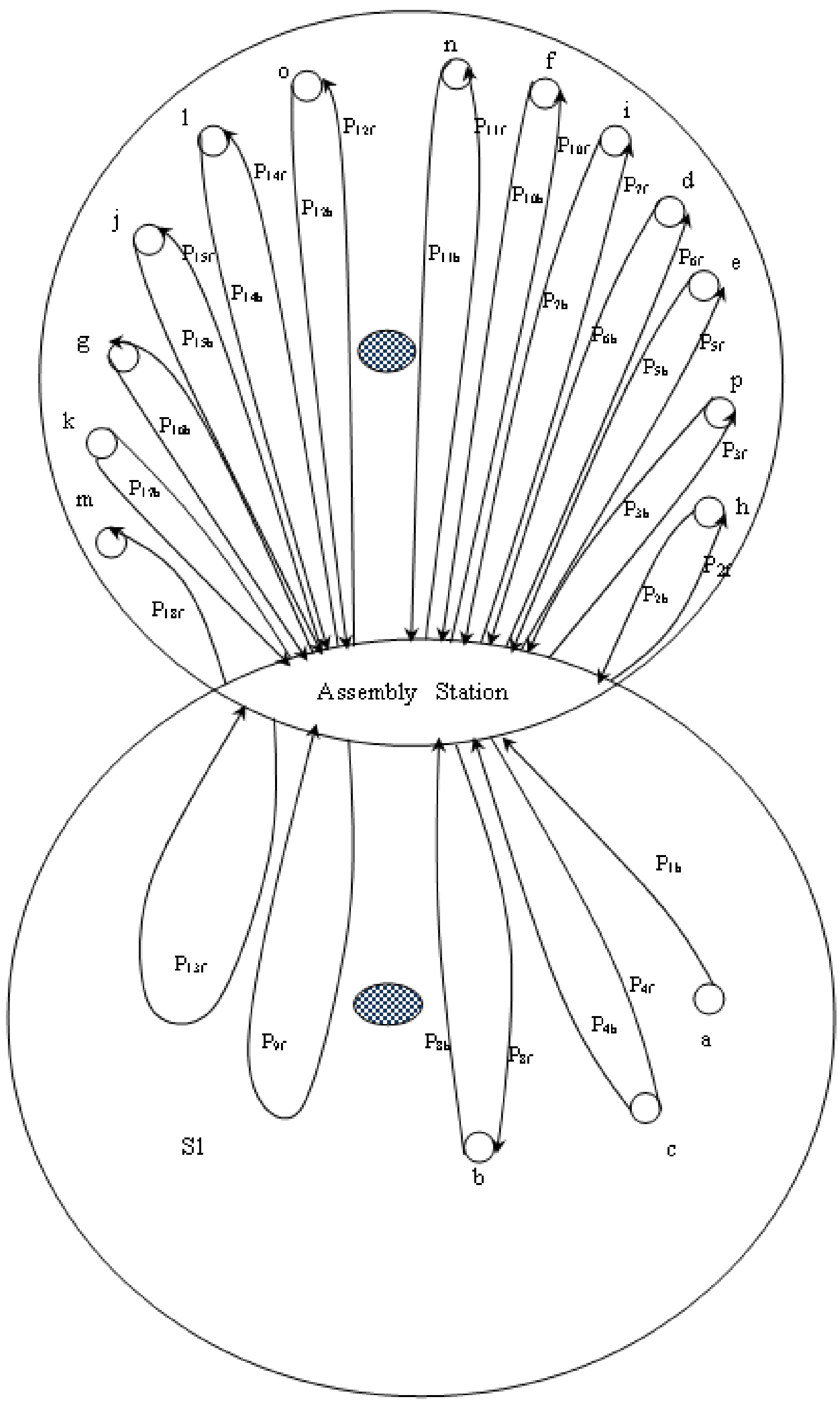


Figure 4.10. Path sequences for driver assembly for both Adept-one and Puma-762 robots

Product-3 (Car Alternator Assembly) for workcell-3

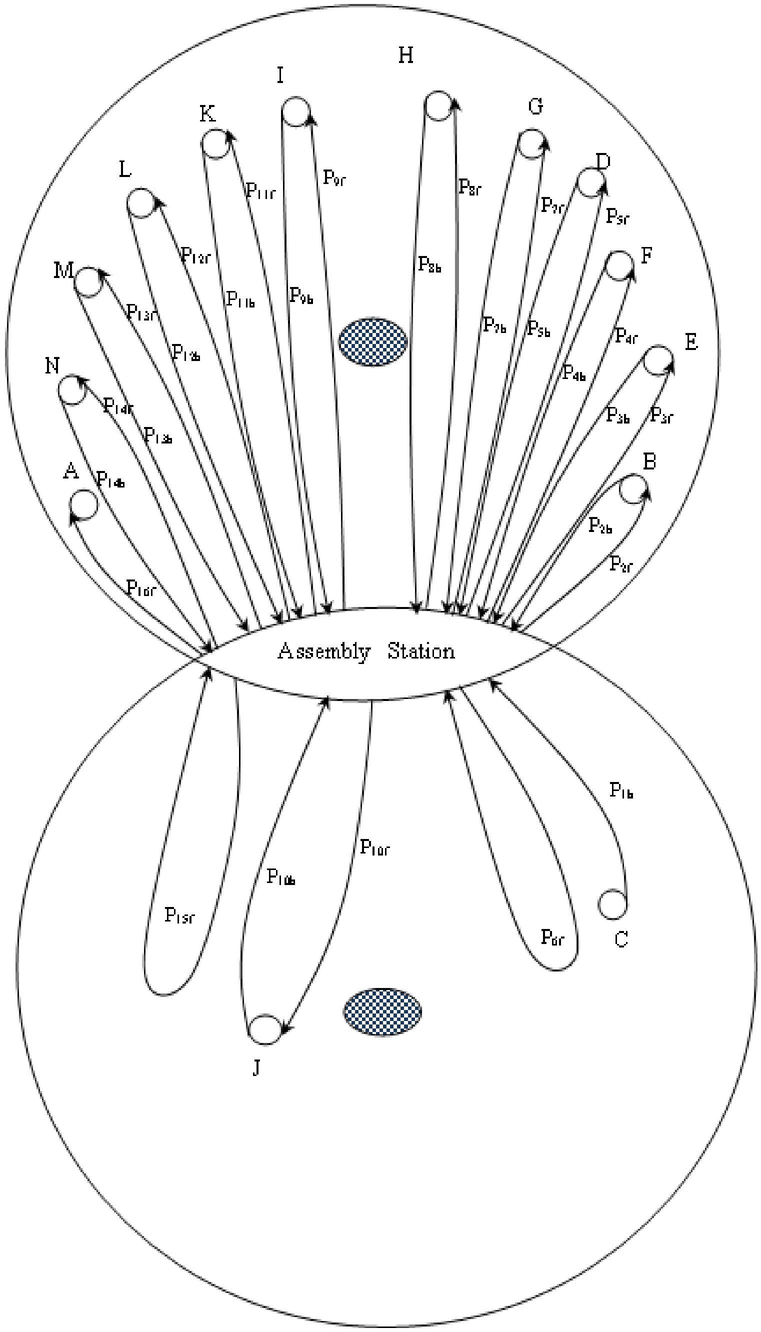


Figure 4.11. Path sequences for car alternator assembly for both Adept-one and Puma-762 robots. Figure 4.9,4.10 and 4.11 shows the path sequences for grinder assembly, driver assembly and car alternator assembly for both Adept-one and Puma-762 robots

4.6 Different paths for different robot:

For Workcell-1:

Table 4.1 Different paths for Adept-one robot

Path Number	Cell	Product	Path	Path Number	Cell	Product	Path
P1101b	1	1	01b	P1208b	1	2	08b
P1102f	1	1	02f	P1209f	1	2	09f
P1102b	1	1	02b	P1209b	1	2	09b
P1103f	1	1	03f	P1210f	1	2	10f
P1103b	1	1	03b	P1210b	1	2	10b
P1104f	1	1	04f	P1211f	1	2	11f
P1104b	1	1	04b	P1211b	1	2	11b
P1105f	1	1	05f	P1212f	1	2	12f
P1201b	1	2	01b	P1212b	1	2	12f
P1202f	1	2	02f	P1213f	1	2	13f
P1202b	1	2	02b	P1213b	1	2	13b
P1203f	1	2	03f	P1214f	1	2	14f
P1203b	1	2	03b	P1214b	1	2	14b
P1204f	1	2	04f	P1215f	1	2	15f
P1204b	1	2	04b	P1215b	1	2	15b
P1205f	1	2	05f	P1216f	1	2	16f
P1205b	1	2	05b	P1301b	1	3	01b
P1206f	1	2	06f	P1302f	1	3	02f
P1206b	1	2	06b	P1302b	1	3	02b
P1207f	1	2	07f	P1303f	1	3	03f
P1207b	1	2	07b	P1303b	1	3	03b
P1208f	1	2	08f	P1304f	1	3	04f
P1304b	1	3	04b	P1309b	1	3	09b
P1305f	1	3	05f	P1310f	1	3	10f
P1305b	1	3	05b	P1310b	1	3	10b

Table 4.1 continue………

P1306f	1	3	06f	P1311f	1	3	11f
P1306b	1	3	06b	P1311b	1	3	11b
P1307f	1	3	07f	P1312f	1	3	12f
P1307b	1	3	07b	P1312b	1	3	12b
P1308f	1	3	08f	P1313f	1	3	13f
P1308b	1	3	08b	P1313b	1	3	13b
P1309f	1	3	09f	P1314f	1	3	14f

For Workcell-2:

Table 4.2 Different paths for Puma-762 robot

Path Number	**Cell**	**Product**	**Path**	**Path Number**	**Cell**	**Product**	**Path**
P2101b	2	1	01b	P2206b	2	2	06b
P2102f	2	1	02f	P2207f	2	2	07f
P2102b	2	1	02b	P2207b	2	2	07b
P2103f	2	1	03f	P22008f	2	2	08f
P2103b	2	1	03b	P2208b	2	2	08b
P2104f	2	1	04f	P2209f	2	2	09f
P2104b	2	1	04b	P2209b	2	2	09b
P2105f	2	1	05f	P2210f	2	2	10f
P2201b	2	2	01b	P2210b	2	2	10b
P2202f	2	2	02f	P2211f	2	2	11f
P2202b	2	2	02b	P2211b	2	2	11b
P2203f	2	2	03f	P2212f	2	2	12f
P2203b	2	2	03b	P2212b	2	2	12f
P2204f	2	2	04f	P2213f	2	2	13f
P2204b	2	2	04b	P2213b	2	2	13b
P2205f	2	2	05f	P2214f	2	2	14f
P2205b	2	2	05b	P2214b	2	2	14b
P2206f	2	2	06f	P2215f	2	2	15f

Table 4.2 continue………

P2215b	2	2	15b	P2307b	2	3	07b
P2216f	2	2	16f	P2308f	2	3	08f
P2301b	2	3	01b	P2308b	2	3	08b
P2302f	2	3	02f	P2309f	2	3	09f
P2302b	2	3	02b	P2309b	2	3	09b
P2303f	2	3	03f	P2310f	2	3	10f
P2303b	2	3	03b	P2310b	2	3	10b
P2304f	2	3	04f	P2311f	2	3	11f
P2304b	2	3	04b	P2311b	2	3	11b
P2305f	2	3	05f	P2312f	2	3	12f
P235b	2	3	05b	P2312b	2	3	12b
P2306f	2	3	06f	P2313f	2	3	13f
P2306b	2	3	06b	P2313b	2	3	13b
P2307f	2	3	07f	P2314f	2	3	14f

<u>For Workcell-3:</u>

Table 4.3 Different paths for both Adept-one and Puma-762 robots

Path Number	**Cell**	**Product**	**Path**	**Path Number**	**Cell**	**Product**	**Path**
P3101b	3	1	01b	P3202f	3	2	02f
P3102f	3	1	02f	P3202b	3	2	02b
P3102b	3	1	02b	P3203f	3	2	03f
P3103f	3	1	03f	P3203b	3	2	03b
P3103b	3	1	03b	P3204f	3	2	04f
P3104f	3	1	04f	P3204b	3	2	04b
P3104b	3	1	04b	P3205f	3	2	05f
P3105f	3	1	05f	P3205b	3	2	05b
P3105b	3	1	05b	P3206f	3	2	06f
P3106f	3	1	06f	P3206b	3	2	06b
P3201b	3	2	01b	P3207f	3	2	07f

Table 4.3 continue………

P3207b	3	2	07b	P3303b	3	3	03b
P3208f	3	2	08f	P3304f	3	3	04f
P3208b	3	2	08b	P3304b	3	3	04b
P3209f	3	2	09f	P3305f	3	3	05f
P3209b	3	2	09b	P3305b	3	3	05b
P3210f	3	2	10f	P3306f	3	3	06f
P3210b	3	2	10b	P3306b	3	3	06b
P3211f	3	2	11f	P3307f	3	3	07f
P3211b	3	2	11b	P3307b	3	3	07b
P3212f	3	2	12f	P3308f	3	3	08f
P3212b	3	2	12f	P3308b	3	3	08b
P3213f	3	2	13f	P3309f	3	3	09f
P3213b	3	2	13b	P3309b	3	3	09b
P3214f	3	2	14f	P3310f	3	3	10f
P3214b	3	2	14b	P3310b	3	3	10b
P3215f	3	2	15f	P3311f	3	3	11f
P3215b	3	2	15b	P3311b	3	3	11b
P3216f	3	2	16f	P3312f	3	3	12f
P3216b	3	2	16b	P3312b	3	3	12b
P3217f	3	2	17f	P3313f	3	3	13f
P3217b	3	2	17b	P3313b	3	3	13b
P3218f	3	2	18f	P3314f	3	3	14f
P3301b	3	3	01b	P3314b	3	3	14b
P3302f	3	3	02f	P3315f	3	3	15f
P3302b	3	3	02b	P3315b	3	3	15b
P3303f	3	3	03f	P3316f	3	3	16f

Table 4.1 and 4.2 shows the different paths for Adept-one or Puma-762 robots and table 4.3 shows the different paths for both Adept-one and Puma-762 robots.

4.7 Summary

A systematic motion planning approach in the workspace of robot has been proposed. It is aimed at enabling robots with capabilities of automatically deciding and executing a sequence of motion in order to achieve a task without collision with other objects in a given environment. It takes into account the configuration space approach for the movement of parts from source to their destination. In this chapter the path sequences for Adept one robot and Puma762 robots has been shown in their workspace. All the path sequences are tabulated. By using conventional motion planning procedure it is difficult task to obtain the best and optimal path. As multiple paths are possible to achieve the objective, it is necessary to select appropriate technique for optimization of paths. Hence the development of a procedure that also accounts for the individual parts along with the subassemblies of the product and places it to their required position such that a safe motion planning can be followed from the source position to goal position with the environment of uncertainties and obstacles .

CHAPTER 5

SOFT COMPUTING TECHNIQUES

5.1 Overview

Many types of optimization tools are available for application to the problem, like Simulated Annealing, Evolutionary Computation, Tabu Search, Ant Colony Optimization, and Artificial Immune System but their suitability and/or effectiveness are also under scanner. Searching the best sequence generation involves the conventional or soft computing methods by following the procedures of search algorithms. Intensification is an expression commonly used for the concentration of search process on areas in search space with good quality solutions. Diversification denotes the action of leaving already exploded areas and moving the search process to unexplored areas. Metaheuristic is set of algorithms concepts that can be used to define heuristic methods applicable to a wide set of different problems.

5.2 Ant Colony Technique

The main concept of ACO is to imitate the cooperative manner of an ant colony to solve combinatorial type's optimization problems within a reasonable amount of time [6]. At the time of their path from nest to food source, ants can deposit and sniff a chemical substance known as pheromone, which provides them with the ability to communicate with each other. An ant lays some pheromone on the ground to mark the path it follows by trail of this substance. Ant move at random, but when they encounter a pheromone trail, they decide whether or not to follow it. The probability that an ant choose one path over others is determined by the amount of pheromone on the potential path of interest. With the continuous action of the colony, the shorter path are more frequently visited and become more attractive for subsequent ants. The main characteristics of an ant algorithm are positive feedback, distributed computation, and the use of a constructive greedy heuristic search. Positive feedback accounts for rapid discovery of good solutions, distributed computation avoids premature convergence, and the greedy heuristic search helps find acceptable solutions in the early stages of the search process. The generic ant algorithms have four main steps as follows:

1. Initialization: Set initial population of the colony and the pheromone trail. Place starting nodes for all ants randomly

2. Solution construction: Taking into account the problem-dependent heuristic info & the trail intensity of the path, each ant choose the next that has been visited to move by probability. Repeat the step till a completed solution is constructed.

3. Trail update: Evaluate the solution and deposit pheromone on the solution paths according to the quality of solution to know about solution whether it is better or not.

4. Pheromone evaporation: The pheromone trail of all paths is decreased by some constant factor at the end of an iteration of building completed solutions.

ACO algorithms have been applied successfully in a variety of optimization problem like Travelling salesman problem, just-in-time sequencing, and job-shop scheduling.

The present research is based on the following assumptions;

i. The possible ant trails joining the nest and food are represented by the possible disassembly sequences of components that, inversely, represent the assembly sequences;
ii. The nest is represented by the first component of the sequence, and the food by the last components;
iii. The concept of trail length (to be minimized) is substituted by the concept of sequence quality (to be maximized), evaluated according to the number of product orientation changes.

5.3 Applying ACO to assembly sequence planning

The motion planning in robotic assembly is much more constrained problem than TSP. An assembly sequence cannot be started from any part, because it may provide unfeasible sequence [6]. For getting a feasible sequence it has to satisfy the geometric, precedence constraints. The basic concept of an ant colony algorithm is to solve combinatorial problems within a reasonable amount of time. Artificial ants iteratively tours through a loop that includes a tour construction biased by the artificial pheromone trails and the heuristic information. The main idea in modified algorithm is that the good tours are the positive feedback given through the pheromone update by the ants. The shorter is the tour the more amounts of pheromones deposits on the selected path. This means that the path have higher probability of being selected in the subsequent iterations of the algorithm. In this study, disassembly sequence is represented as disassembly operations (DO). The sequence considered the number of parts presented and the direction in which it is to be disassembled i.e. DO = (n, d), where ‘n’ is the number of components and ‘d’ is the direction of disassembly. In this paper, each component is having five possible DOs, i.e. (n,+x), (n, +y), (n, +z), (n, -x) and (n, -y). If the assembly consists of ‘n’ number of parts, then the disassembly operation is having ‘5n’ number of nodes. The disassembly operation is assigned to ‘1’ if there is interference in that direction, otherwise ‘0’. That means if DO=1, it cannot be disassembled from the product. In the modified ACO algorithms, a pheromone ‘ τ_{ij} ’ is used as the share memory of all ants and simultaneously it considers the energy matrix which is to be minimized. The pheromone ‘ τ_{ij} ’ is updated during the processing. In this study the pheromone is expressed as 5n X 5n matrix as because one of the Z directions is restricted in study. The interference matrix in (+) ve X, Y, Z directions is given as;

$$DM = \begin{array}{c} \\ e_1 \\ e_2 \\ \cdots\cdots \\ e_n \end{array} \!\!\! \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & & e_3 \end{array} \\ \begin{bmatrix} I_{11x}I_{11y}I_{11z} & I_{12x}I_{12y}I_{12z} & \cdots\cdots & I_{1nx}I_{1ny}I_{1nz} \\ I_{21x}I_{21y}I_{21z} & I_{22x}I_{22y}I_{22z} & \cdots\cdots & I_{2nx}I_{2ny}I_{2nz} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ I_{n1x}I_{n1y}I_{n1z} & I_{n2x}I_{n2y}I_{n2z} & \cdots\cdots & I_{nnx}\,I_{nny}I_{nnz} \end{bmatrix} \end{array} \tag{5.1}$$

Where I_{ijd} is equal to 1 if component e_i interferes with the component e_j during the move along direction +d-axis; otherwise I_{ijd} is equal to 0. The initial disassembly matrix is calculated as:

$$DM = \begin{array}{c} \\ \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccccccccccccccccccccccc} & & a & & & & & b & & & & & c & & & & & d & & & & & e & & \\ x & y & z & \bar{x} & \bar{y} & x & y & z & \bar{x} & \bar{y} & x & y & z & \bar{x} & \bar{y} & x & y & z & \bar{x} & \bar{y} & x & y & z & \bar{x} & \bar{y} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \tag{5.2}$$

$$DO_{i,(+d)} = \bigcup_{j=1}^{n} I_{ijd} \tag{5.3}$$

$$DO_{i,(-d)} = \bigcup_{j=1}^{n} I_{jid} \tag{5.4}$$

$$\begin{bmatrix} DO_{a,x}=1 & DO_{a,y}=1 & DO_{a,z}=1 & DO_{a,\bar{x}}=1 & DO_{a,\bar{x}}=1 \\ DO_{b,x}=1 & DO_{b,y}=1 & DO_{b,z}=1 & DO_{b,\bar{x}}=1 & DO_{b,\bar{y}}=1 \\ DO_{c,x}=1 & DO_{c,y}=1 & DO_{c,z}=1 & DO_{c,\bar{x}}=1 & DO_{c,\bar{y}}=1 \\ DO_{d,x}=1 & DO_{d,y}=1 & DO_{d,z}=1 & DO_{d,\bar{x}}=1 & DO_{d,\bar{y}}=1 \\ DO_{e,x}=1 & DO_{e,y}=1 & DO_{e,z}=1 & DO_{e,\bar{x}}=1 & DO_{e,\bar{y}}=1 \end{bmatrix} \tag{5.5}$$

Here, U is the Boolean operator OR. The result will be equal to 0 if all the elements involving in the operation are 0. This means the element can be disassembled in that direction. If the DO is equal to 1, the element cannot be disassembled. In this study, the initial feasible disassembly

operations are: (c, -x) and (e, +x).

5.4 The solution:

Motion planning in robotic assembly is a case of combinatorial optimization problem [7]. The problem is similar to Traveling salesman problem i.e. to give the shortest path with minimum cost. Combinatorial optimization problem is a triple (S, f, Ω), where S is the set of candidate solutions, f is the objective function which assigns an objective function value $f(s)$ to each candidate solution s $\in S$, and Ω is a set of constraints. The solutions belonging to the set of solutions S that satisfies the constraints Ω are called feasible solutions. The stable solutions $\tilde{\Omega} \subseteq \Omega$ belong to the feasible solutions. One of the major advantages is that, the optimal solution satisfies all the assembly constraints, objective function and also it is a part of stable solutions $\tilde{\Omega}$.

In ant system, m ants simultaneously build a solution of the ASG. Initially ants are put in first feasible DO. At each construction step, ant k applies a probabilistic state transition rule, called *random proportional rule,* to decide which node visit next.

$$P_d = \begin{cases} \dfrac{[\tau(i,j)]^{\alpha}[\eta(i,j)]^{\beta}}{\sum\limits_{u \in C_d(i)} [\tau(i,j)]^{\alpha}[\eta(i,j)]^{\beta}}, if \;\; j \in C_d \\ 0, \qquad\qquad\qquad\qquad otherwise \end{cases} \tag{5.6}$$

The heuristic value selected in this study is $\eta(i,j) = \dfrac{1}{Time}$

After all the ants have constructed their tours, the pheromone trails are updated. The pheromone evaporation is giving by $\tau(i, j) \leftarrow (1-\rho)\tau(i, j)$ where $0 \le \rho \le 1$ is the pheromone evaporation rate. After evaporation, all ants deposit pheromone on the arcs they have crossed in their tour:

$$\tau(i, j) \leftarrow (1-\rho)\tau(i, j) + \sum_{k=1}^{m} \Delta\tau_k (i, j) \tag{5.7}$$

Where *m* is the number of ants that find the iteration-best sequences and $\Delta\tau^k(i, j)$ is the amount of pheromone ant *k* deposits on the arcs it has visited. It is given an equation:

$$\Delta\tau^k (i, j) = \begin{cases} \dfrac{1}{Time(i, j)}, & if\,(i, j) \in sequence\, of\ ant\, k \\ 0, & otherwise \end{cases} \tag{5.8}$$

where, *Time(i, j)* is time taken by the k^{th} ant belonging to that tour. During the construction of sequences, local pheromone updating encourages exploration of alternative solutions, while global pheromone updating encourages exploitation of the most promising solutions.

5.5 ACO Algorithm

1. Generate the initial feasible DOs and compute their quantity
2. Set the cycle counter NC = 1
3. While NC < NC_{max}
   a. Place ants on the initial feasible nodes of the DCG
   b. While each ant has not completed its tour
      i. Put current DO into sequence of the ant
      ii. Generate candidate list of the ant and calculate the time
      iii. Calculate $p_k(i, j)$ of each candidate
      iv. Choose next DO j based on energy matrix
      v. Move the ant to DO j
      vi. Add the component number of DO j to the tabu list of the ant
      vii. Locally update PM
   c. Evaluate all solutions taking into account their reorientations

d. Globally update PM using iteration-best solutions

e. Update the best sequence of each ant if its iteration sequence is the best one found so far

f. Empty the sequence, candidate list, and tabu list of each ant

g. Set *NC=NC+1*

4. Output the reversed best sequence of each ant

The reversed best sequence of each ant is listed. The reverse of the output is the optimal assembly sequence with inverse directions. The solution is either optimal or near optimal. The flowchart of the ACO procedure is presented in Figure 5.1.

Flowchart for the ACO :-

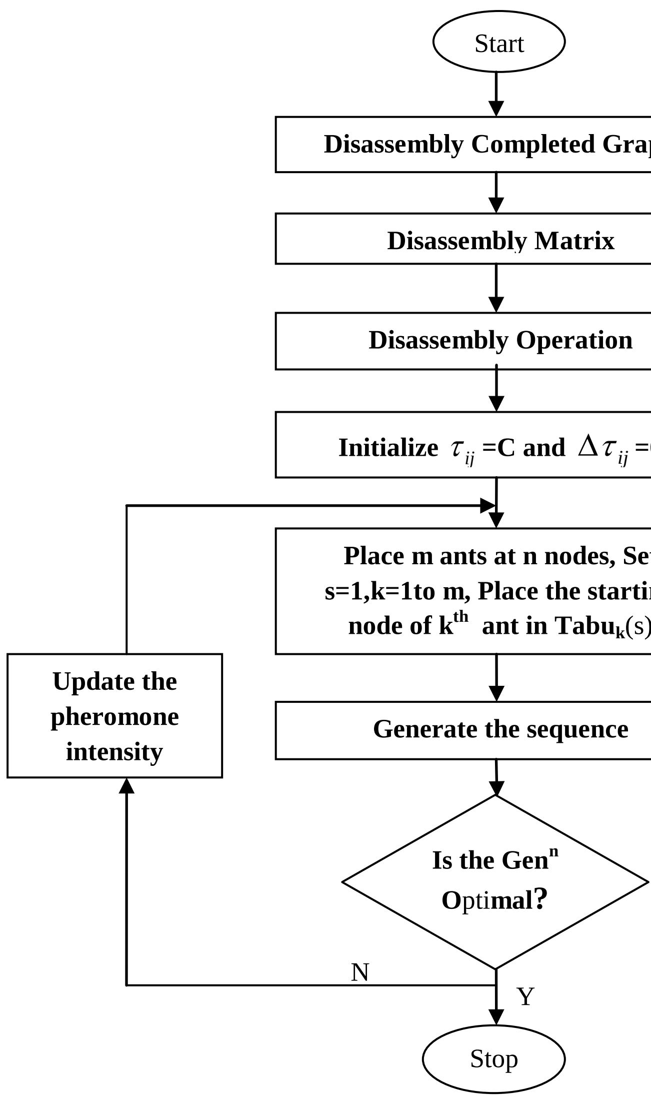


Figure 5.1.The flowchart of the ACO algorithm

5.6 Summary

The soft computing technique carried out in this chapter generates the stable motion planning sequence following all the constraints and optimize the stable sequences to give out the best result. A motion planning sequence is considered to be optimal when it minimizes travelling time while satisfying the process constraints. Here the work utilizes an ant colony optimization (ACO) method for generation of optimal path sequence. The travelling time is minimized by ACO. Example problems are presented to show the effectiveness of the method. This modified method will be more suitable in path sequence that considers all the constraints and travelling time.

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Overview

The result obtained by using Ant Colony Optimization (ACO) method for the products under consideration is presented in the following section. It is one of the effective metaheuristic optimization tool used to solve the robotic assembly sequence generation and motion planning sequence. The main characteristics of ant colony algorithms are positive feedback, distributed computation, and the use of a constructive greedy heuristic search. The ant's behavior, their principles and mechanisms of methodology are used to solve the problem of robotic assembly and motion planning sequence generation. Three example problems are chosen to test the developed technique. The following sections present the results obtained through ACO method and the related discussions.

6.2 Path sequences

The possible assembly sequences and path sequences for product-1 are

1. a-b-d-c-e
2. a-b-d-e-c
3. a-b-c-d-e
4. a-d-b-c-e
5. a-d-b-e-c
6. a-d-e-b-c
7. d-a-e-b-c
8. d-a-b-c-e
9. d-a-b-e-c
10. c-b-a-d-e
11. b-a-d-e-c
12. b-a-d-c-e
13. b-a-c-d-e

The possible assembly sequences and path sequences for product-2 are

1. a-h-p-c-e-d-i-b-f-n-o-l-j-g-k-m
2. f-n-o-l-j-g-k-m-b- a-h-p-c-e-d-i
3. a-b-e-c-h-p-i-d-f-l-j-k-g-m-n-o
4. f-n-o-l-j-g-k-m- e-d-i-b- a-h-p-c
5. a-h-p-c-e- l-j-g-k-m-d-i-b-f-n-o
6. n-f-o-m-g-h-i-p-d-e-a-b-c-k-l-j
7. o-n-m-g-h-p-i-d-f-l-j-k-a-b-e-c
8. o-m-n-g-h-p-i-d-f-j-k-l-a-b-e-c
9. a-h-p-b-c-d-e-k-l-j-p-f-g-m-n-o
10. a-h-p-c-e-d-i-b-f-o-n-l-j-g-k-m
11. a-h-p-b-c-e-d-i-f-n-o-l-j-g-k-m
12. h-i-p-m-g-f-n-o-d-e-b-c-a-k-l-j
13. a-h-i-p-c-b-e-d-k-l-j-g-m-f-n-o
14. a-h-i-p-c-e-d-b-f-n-o-l-j-g-k-m
15. a-h-p-c-e-e-d-i-b-f-n-o-k-j-l-g-m

The possible assembly sequences and path sequences for product-3 are

1. C-B-E-F-D-G-H-I-J-K-L-M-N-A
2. C-E-F-D-G-H-I-J-K-L-M-N-B-A
3. A- C-E-F-D-G-H-I-J-K-L-M-N-B
4. J-K-L-M-N-A- G-H-I- C-B-E-F-D
5. J-K-L-M-N- G-H-I- C-E-F-D- B-A
6. B-C-E-F-D-G-H-I-J-K-L-M-N-B-A
7. A-N-M-L-K-J-I-H-G-D-F-E-C-B
8. J-K-L-M-N-A-C-B-D-E-F-G-H-I
9. J-K-L-M-N-A-C-B-E-F-D-G-H-I
10. G-H-I-C-E-F-D-J-K-L-M-N-B-A
11. B-C-D-E-F-G-H-I-J-K-L-M-N-A
12. M-N-A-C-B-E-F-D-G-H-I-J-K-L

6.3 Results and discussions

An ant colony based approach has been used to generate optimal stable assembly sequence and then the optimal path sequence. The conventional methods like liaison method, connectivity graph method, matrix method, disassembly methods give multiple solutions. As the number of parts increases in assembly products these methods provide multiple sequence and getting the optimal sequence is quite troublesome. However ant colony optimization method is one of the metaheuristic methods to solve these types of combinatorial optimization problems. It has been observed that, lower value of pheromone allow a fast convergence toward the solution. The lower value accelerates the evaporation process of the pheromone in low quality trails, increasing more and more the relevance to get the best solutions [7]. The ant colony algorithm starts from searching the first disassembly node to the last one. In between the search process follows the path based on the kind of parameters selected and to an extent pheromone used. Ultimately, the sequence generated in the algorithm is the optimal disassembly sequence to that product. The reverse of it is the optimal assembly sequence.

The work considered three examples to measure the accuracy of algorithm and the following results are obtained.

1. In grinder assembly the optimal path sequence is: **a-d-e-b- c**

Fig.6.1.shows the optimal sequence for grinder assembly for workcell-1 and workcell-2

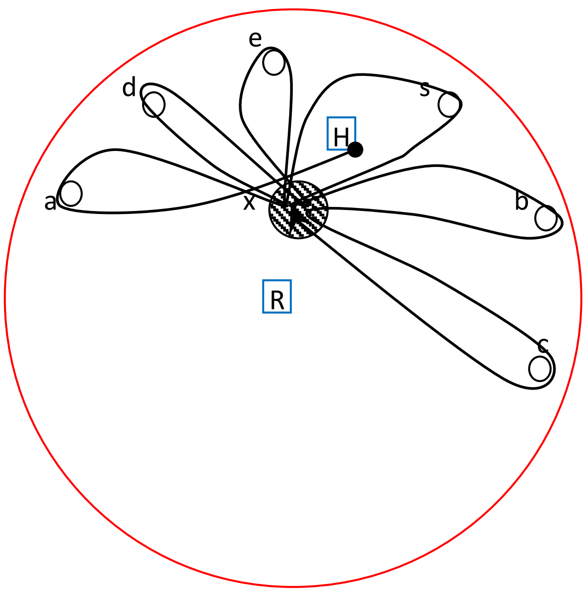


Figure 6.1.The optimal sequence for grinder assembly for workcell-1 and workcell-2

Table 6.1 shows task decomposition for optimal sequence for grinder assembly for work cell-1 and work cell-2 and workcell-3

Table 6.1.Task decomposition for optimal sequence for grinder assembly for work cell-1 and work cell-2 and workcell-3

Sl no.	Part Name	Part ID	Task
1	Shaft	a	Pick-rotate- orient -move-place
2	Blade	d	Pick- rotate-move-place
3	Nut	e	Pick -move-insert-place
4	Sub assembly-1	a-d-e	Pick-rotate-orient-move-place
5	Blade	b	Pick- rotate-move-place
6	Nut	c	Pick- move-insert-place

Fig.6.2. shows the optimal sequence for grinder assembly for workcell-3

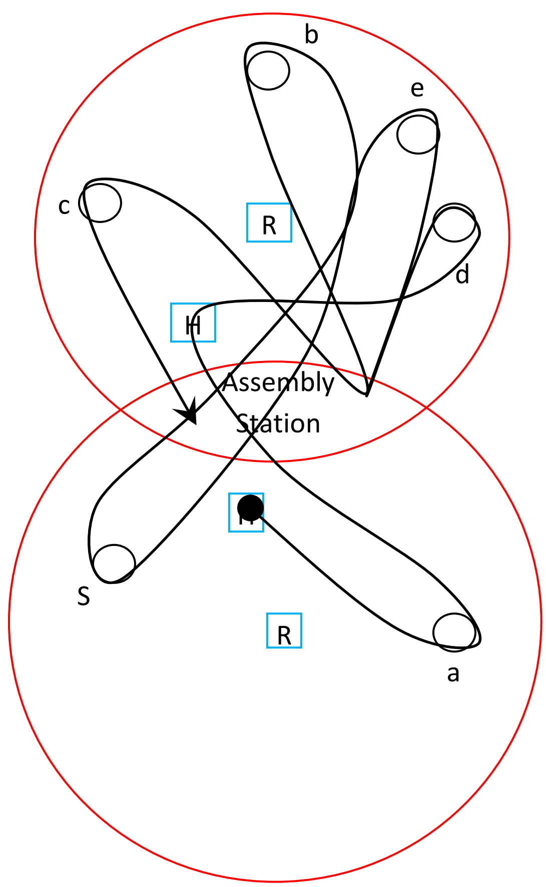


Figure 6.2.The optimal sequence for grinder assembly for workcell-3

2. In driver assembly the optimal path sequence is: **a-h-p-c-e-d-i-b-f-n-o-l-j-g-k-m**

Table 6.2 Task decomposition for optimal sequence of driver assembly for work cell-1 and work cell-2 and workcell-3

Sl no.	**Part**	**Part ID**	**Task**	**Sl no.**	**Part**	**Part ID**	**Task**
1	Base	a	Pick- rotate-move-place	10	Cover	f	Pick-rotate-move-place
2	Electro-motor	h	Pick-rotate-move-place	11	Screw	n	Pick- move-insert-place
3	Screw	p	Pick- move-insert-place	12	Screw	o	Pick- move-insert-place
4	Bush2	c	Pick- move-insert-place	13	Sub-assembly-2	f-n-o	Pick- rotate-orient-move-place
5	Pillar2	e	Pick- move-insert-place	14	Setup Screw	l	Pick- move-insert-place
6	Pillar1	d	Pick- move-insert-place	15	Shell	j	Pick- move-insert-place
7	Sensor	i	Pick- move-insert-place	16	Plug	g	Pick-rotate-move-insert-place
8	Bush1	b	Pick- move-insert-place	17	Screw	k	Pick- move-insert-place
9	Sub-assembly1	a-h-p-c-e-d-i-b	Pick- rotate-orient-move-place	18	Screw	m	Pick- move-insert-place

Table 6.2 shows the task decomposition for optimal sequence of driver assembly for work cell-1 and work cell-2 and workcell-3

Fig.6.3. shows the optimal sequence for driver assembly for workcell-1 and workcell-2

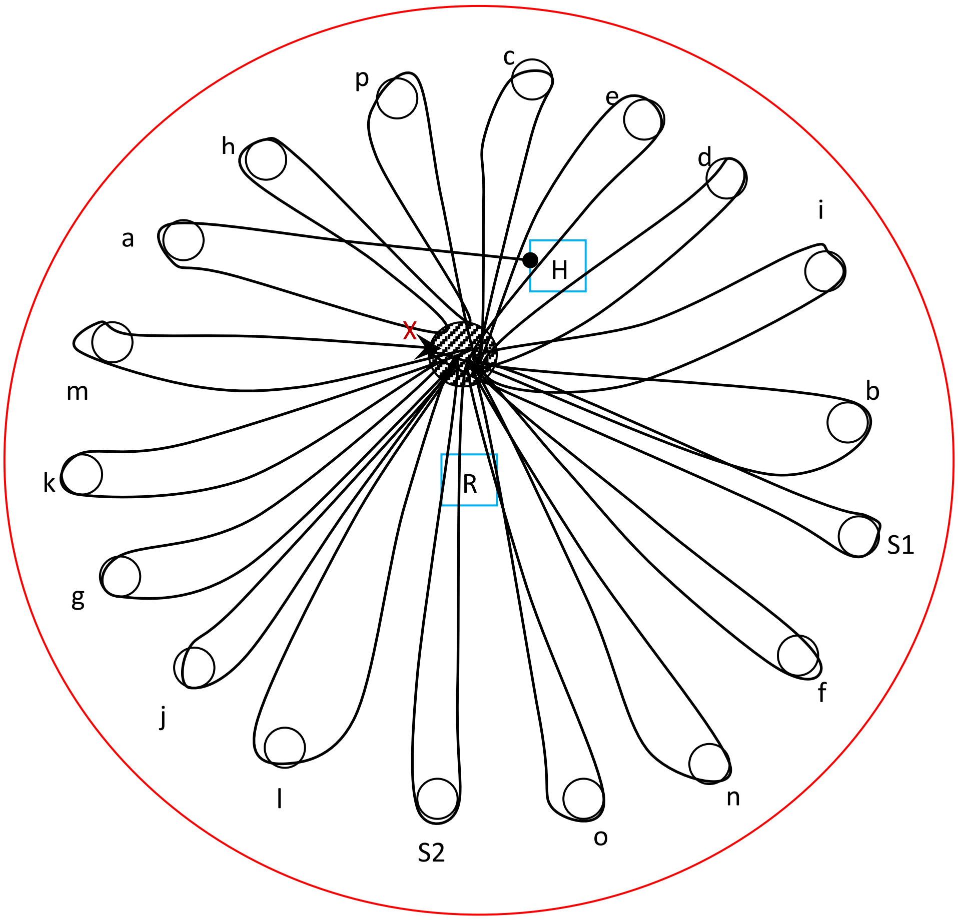


Figure 6.3. The optimal sequence for driver assembly for workcell-1 and workcell-2

Here X → Assembly Station

H → Robot Home Position

R → Robot Base Position

Figure 6.4. shows The optimal sequence for driver assembly for workcell-3

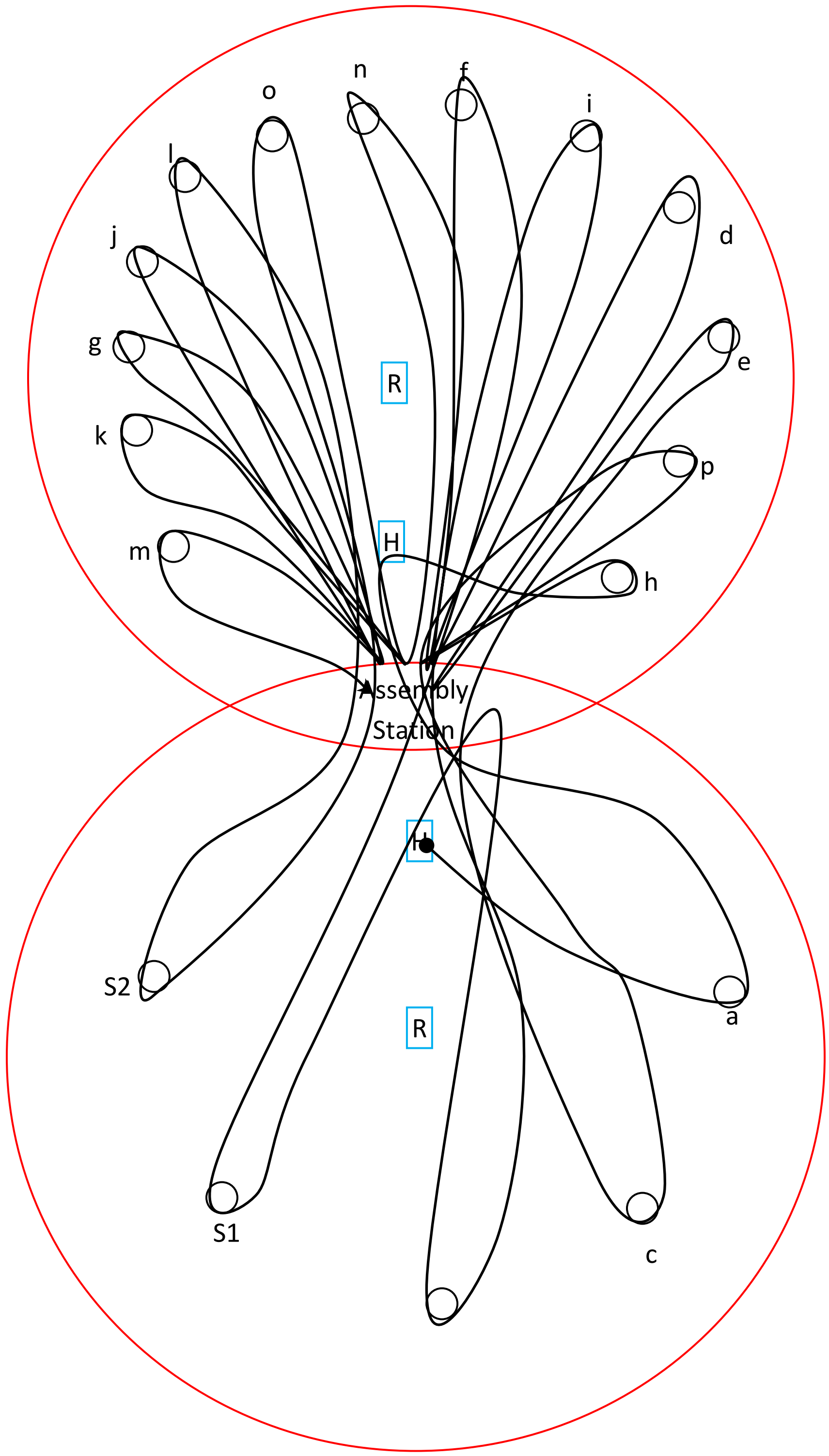


Figure 6.4. The optimal sequence for driver assembly for workcell-3

3. In car alternator assembly the optimal path sequence is: **C-B-E-F-D-G-H-I-J-K-L-M-N-A**

Table 6.3: Task decomposition for optimal sequence of car alternator assembly for work cell-1 and work cell-2 and workcell-3

Sl no.	**Part Name**	**Part ID**	**Task**	**Sl no.**	**Part Name**	**Part ID**	**Task**
1	Drive Frame	C	Pick-rotate-move-place	9	B2 cover	I	Pick- move- insert-place
2	Space collar	B	Pick-rotate-move-insert-place	10	Rear frame	J	Pick- move-insert-place
3	Bearing 1	E	Pick-rotate-move-insert-place	11	Rectifier	K	Pick- rotate-move-attach-place
4	Retainer	F	Pick-rotate-move-insert-place	12	IC Regulator	L	Pick- rotate-move-attach-place
5	Stator	D	Pick-rotate- move -insert-place	13	Brush & Holder	M	Pick- move-insert-place
6	Sub-assembly-1	C-B-E-F-D	Pick-rotate-orient-move-place	14	Rear cover	N	Pick-rotate-move-place
7	Rotor	G	Pick-rotate-move-place	15	Sub-assembly-2	G-H-I-J-K-L-M-N	Pick-rotate-orient-move-place
8	Bearing 2	H	Pick- move- insert-place	16	Pulley	A	Pick-rotate- move-insert-place

Table 6.3 shows the task decomposition for optimal sequence of car alternator assembly for work cell-1 and work cell-2 and workcell-3

Figure 6.5. shows the optimal sequence for car alternator assembly for workcell-1 and workcell-2

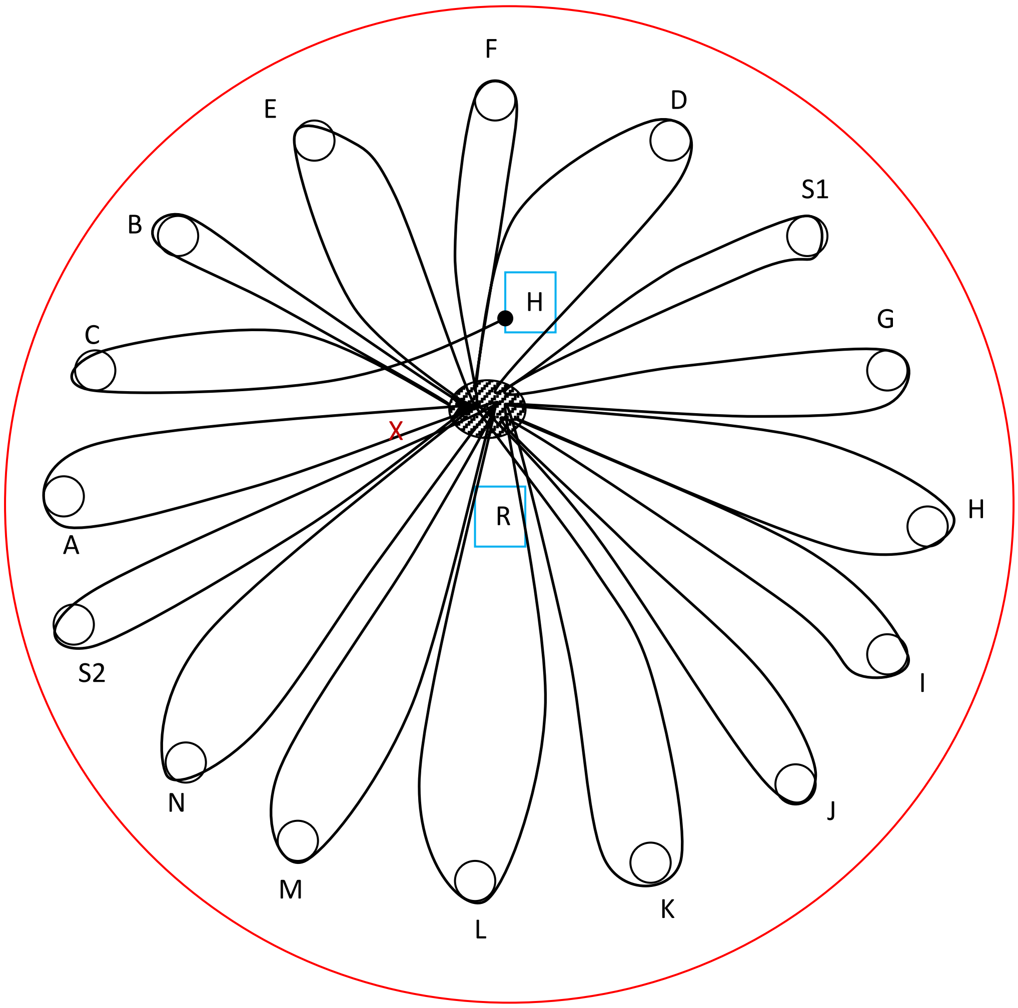


Figure 6.5.The optimal sequence for car alternator assembly for workcell-1 and workcell-2

Figure 6.6. shows the optimal sequence for car alternator assembly for workcell-3

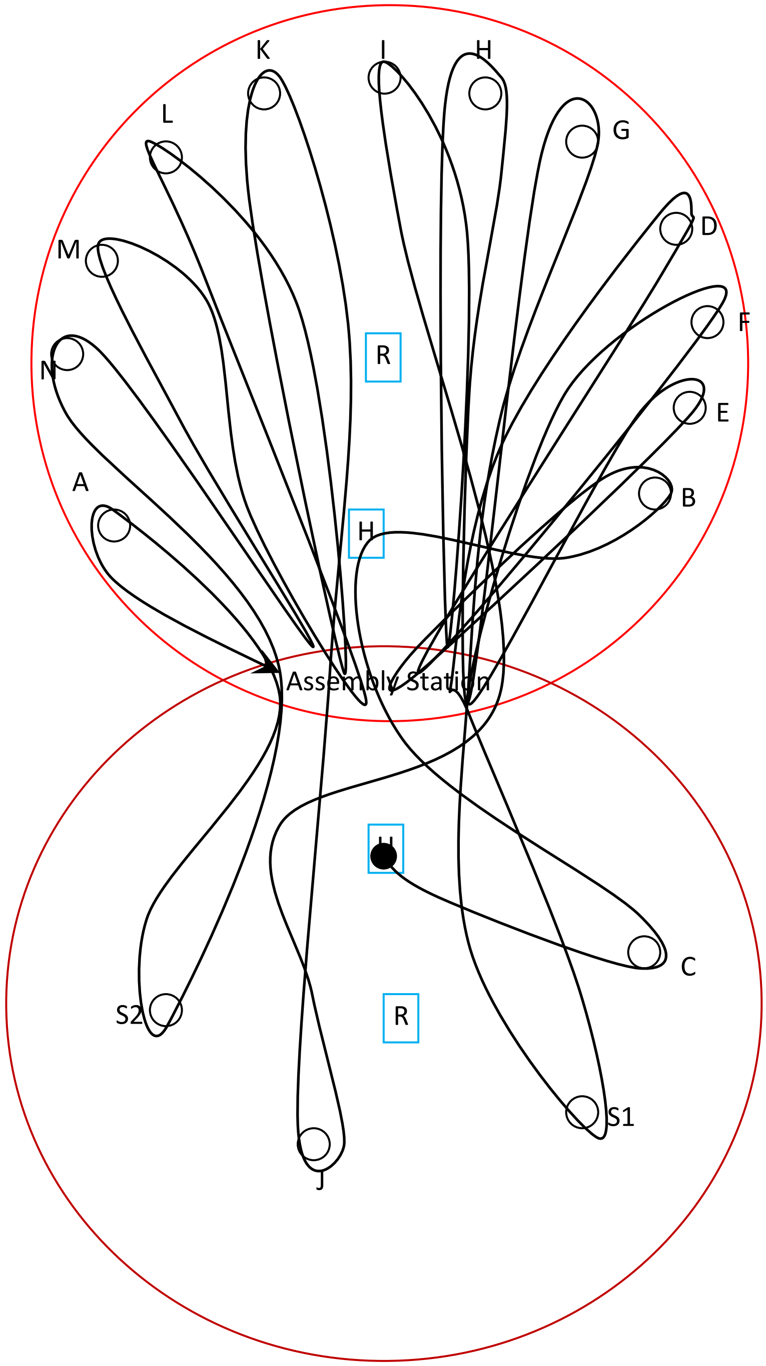


Figure 6.6.The optimal sequence for car alternator assembly for workcell-3

6.4 Summary

The optimized path for grinder assembly, driver assembly and car alternator assembly are developed using ACO techniques .The results are shown and described in the above figures.

CHAPTER-7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

A basic motion planning of a robot is a process to produce a continuous motion that connects a start configuration ‘S’ and a goal configuration ‘G’ avoiding collision with obstacles. The robot and obstacle geometry are described in a 2D or 3D workspace. Robots pick and place parts in assembly sequence using a predetermined pattern of movement and hence produce the path sequence. Until recently, much effort has been devoted in safe motion planning in the presence of obstacles and uncertainties. Path planning in the workspace of robot for a product assembly depends on the assembly of parts of the product. If the number of parts of a robotic assembly increases, the sequence of parts in a product becomes complicated and hence it is difficult to make path sequences in between the parts in the robot workspace. As multiple no. of paths are available in the robot workspace of a product assembly, by applying conventional methods it is quite a difficult task to optimize the path sequence. Since it is a type of combinatorial optimization problem, it is more suitable to use metaheuristic method to optimize the path.

The metaheuristic method presented here is ACO technique because of the following advantages.

- It can be used in multi objective function.
- It is easy to understand.
- It is used to minimize the lead time and work in process time.
- It computes shortest path easily.
- It can solve large problems in short period of time.
- Its efficiency and performance is more.

The work presents the approach for the generation of path sequence, testing feasibility of the sequence and finding the optimal sequence minimizing the travel time and hence provides a new dimension to this subject. In summary, the present work can be seen as a guideline for many researchers to make safe path planning in between the parts of a robotic assembly in the robot workspace.

The work conducted during this project may be summarized as follows:

- As this work is related to assembly product different products are chosen and then assembly product is disassembled to different parts.
- The assembly planning procedure is developed following precedence constraints, geometric constraints and connectivity constraints.
- The industrial robots are selected according to the tasks to be carried out and weight, shape and size of the parts to be handled.
- Tasks are allocated to robots and motion planning is done and all the feasible sequences are developed.
- Depending upon the number of parts in the products and their manipulation requirements, multiple paths are recorded.
- Applying appropriate optimization technique i.e. ACO to all these feasible paths, the optimal path is determined.

The result of this work have been compared with that obtained by previous researchers [6,7,10] and it has been found that the present method is quite effective and faster.

7.2 Future Scope of work

Through extensive research works have been carried out, several areas for future research still remain open. Several method for the generation of safe motion planning of robot are discussed in course of the present work. However, the work concentrated on important and simple method for selecting appropriate method for path sequence. Nevertheless, sufficient research outcome may be realized some of the following areas such as;

1. This motion planning problem can also be solved by applying other techniques like particle swarm optimization (PSO), artificial immune system (AIS).
2. It can also be verified by comparing with the above techniques to get the best result.
3. A single computer program can also be developed which integrates assembly sequence generation, path planning, path optimization and robot programming to get the best result

REFERENCES

[1] D Fazio T. and Whitney D., "Simplified generation of all mechanical assembly sequences", *IEEE J. Robotics and Automations* RA-3, No. 6, 1987, pp. 640-658

[2] De Fazio and Whitney,"An Integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Products", *Robotics and Automation, IEEE Transactions*, VOL. ***7.*** NO. 1.1991, pp. 78-9413

[3] Chan P., "Automatics assembly sequence generation by pattern matching", *IEEE Transaction on Systems, Man and Cybanetics*, 1991, pp. 376-389

[4]de Mello.L.S.H., Sanderson.A.C., "A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences", *Robotics and Automation, IEEE Transactions*, VOL. ***7.*** NO. 2.1991, pp.228-240

[5] Mascle. C, Jabbow. T & Maramam.R, "ASSEMBLY FEATURES FOR MECHANICAL PRODUCT DATA", *International Symposium on Assembly and Task Planning,IEEE*,1997, pp. 218-223

[6] Sharma.S., Mohapatra.R.N., Biswal. B.B, Choudhury B.B.," Generation of Robotic Assembly Sequence using Ant Colony Optimization", *Fourth International Conference on Industrial and Information Systems, ICIIS*, 2009, pp.520-525

[7] Wang J.F., Liu J.H. and Zhong. Y.F., "A novel ant colony algorithm for assembly sequence planning " , *Int J Adv Manuf Technol* (2005**)** 25, pp.1137–1143

[8] Dorigo. M.," Ant colony system: A cooperative learning approach to the travelling salesman problem", *IEEE Transactions on Evolutionary Computation. IEEE*,1(1), 1997, pp.53-66

[9] Dorigo M et al., “The ant system: optimization by a colony of cooperating agents”, *IEEE Trans Syst Man Cybernet*, 26(1), 1996, pp. 29–41.

[10] Fujimoto. H. and Sebaaly. M.F.," A New Sequence Evolution Approach to Assembly Planning ", *Transactions of the ASME*, Vol. 122, FEBRUARY 2000, pp.198-205

[11] Lozano-Perez. T. and. Wesley.M. A," An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles", *Communications the ACM,* volume 22, October 1979, pp. 560-570

[12] Xiao. J and Volz. R.A, "On Replanning for Assembly Task Using Robots in the Presence of Uncertainties",1989 ,*IEEE*, pp. 638-645

[13] Singh.S.K., "Trajectory Planning With Uncertainty in Manipulators", *IEEE J. Robotics and Automations*, 1989, pp.152-158

[14] Shibata.M. and Ohnishi.K, "Collision Detection and Distance Calculation For Safe Path Planning Based on Mathematical Programming Methods", *Proc.of Int"l Conf. of IEEE*, 1992, pp. 832-837

[15] Shao.K and Young.K.Y., "Path Feasibility and Modification Based on the PUMA 560 Workspace Analysis*", Proc. of.30th Conf. of IEEE*, 1994, pp. 36-43

[16] Wong.A.K.C., Mayorga. R.V., Rong.L., and Liang. X, "A VISION BASED ON-LINE MOTION PLANNING OF ROBOT MANIPULATORS*," Proc. IROS, IEEE*,1996, pp.940-948

[17] Sutanto. H and Sharma. R, "Practical Motion Planning with Visual Constraints", *International Symposium on Assembly and Task Planning*, *Proceedings of the 1997 IEEE*, pp. 237-242

[18] Hsu.D., Latombe. J.C., and Sorkin.S.,"Placing a Robot Manipulator Amid Obstacles for Optimized Execution", *Assembly and Task planning, Proc. Int'l Symp. of IEEE*, 1999, pp. 1-6

[19] Sharma. R, LaValle. S. M., Seth A.,"Optimizing Robot Motion Strategies for Assembly with Stochastic Models of the Assembly Process", *Assembly and Task Planning, Int'l Symp. of IEEE,* 1995**,** pp.341-346

[20] Divelbiss.A.W and Wen. J. T.,"A path Space Approach to Nonholonomic Motion Planning In the Presence of Obstacles", *IEEE Transactions on ROBOTICS AND AUTOMATION*,vol.13,No.3,JUNE **1997**, pp.443-450

[21] Yue. S., Henrich. D., Xu W.L. and Tso. S. K.," Point-to-Point trajectory planning of flexible redundant robot manipulators using genetic algorithms", ,*Robotica* (2002) volume 20, pp. 269–280.

[22] Pires. E. J.S., Machado. J.A.T. and Oliveira.P.B.M., " Fractional Order Dynamics in a Genetic Algorithm", *the 11th International Conference on Advanced Robotics,* 2003, pp. 264-269

[23] Miura J. and Shirai Y., "Modeling Motion Uncertainty of Moving obstacles for Robot Motion Planning", Proceedings of the 2000 IEEE International Conference on Robotics & Automation, pp.2258-2263

[24] Kazem. B.I., Mahdi. A .I. and Oudah. A. T.," Motion Planning for a Robot Arm by Using Genetic Algorithm", *Jordan Journal of Mechanical and Industrial Engineering,* Volume 2, Number 3,Sep. 2008, pp.131-136

[25] Zhao.C.S, Farooq.M., Bayoumi.M.M., "Kinematic Motion Constraints in the Robot's Joint Space", *Proc. of 22nd Int'l Conf. of IEEE***,** 1996, pp.1246-1251

[26] Jamisola. R. S., Jr.&Anthony A. Maciejewski. A. A., Roberts. R. G., "A Path Planning Strategy for Kinematically Redundant Manipulators Anticipating Joint Failures in the Presence of obstacles", *Intelligent Robots and Systems, Proc. of Int'l Conf. IEEE*, 2003, pp. 142-148

[27] Cho and Cho H., " Interference on robotic assembly precedence constraints using part contact level graph", *Robotica*, Vol. 11, 1993, pp.351-362

[28] Schilling R.J, Book, "Fundamentals of Robotics", Prentice -Hall of India, 1996, pp. 357-394